



UNIVERSITAT JAUME I

**ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES EXPERIMENTALS
GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES**

***Automatización de la entrada de camiones
en una industria***

TRABAJO FIN DE GRADO

AUTOR/A

Pascual Mezquita Parra

DIRECTOR/A

Pedro Balaguer Herrero

Castellón, Noviembre de 2020

Agradecimientos.

Primeramente, me gustaría agradecer el apoyo recibido por parte de mi familia por el sacrificio que han tenido que hacer para que pueda realizar estos estudios y la ayuda mostrada todos los días dándome fuerza en los momentos difíciles. También agradecer a mis amigos por su ayuda también ha servido para levantarse en esos momentos complicados.

Por otro lado, quería agradecer a mi tutor, Pedro Balaguer Herrero, por su apoyo y consejos en todas las tutorías realizadas durante todo este tiempo que me han servido para poder realizar este trabajo y además aprender más sobre el campo de la automatización.

También quería agradecer a la empresa Mas Vicent Ingenieros S.L. por darme la oportunidad de poder realizar las practicas, formándome dentro del ámbito de la ingeniería, y proponerme la realización de este Trabajo Final de Grado. También agradecerles el apoyo y la ayuda recibida para poder realizarlo correctamente con todas las dudas que han surgido sobre la realización.

Por último, agradecer a todos los profesores y compañeros que me he encontrado durante la realización de todo el grado porque siempre han estado allí para ayudarme y solucionarme todo lo que pudieran.



Índice general.

Memoria.....	11
Anexos.....	59
Pliego de condiciones.....	107
Presupuesto.....	117



Índice de imágenes del proyecto.

Imagen 1. Diagrama de Venn de 3 conjuntos que representa la idea general del proyecto.	15
Imagen 2. Representación visual de la función que realiza Node-RED en este proyecto.	16
Imagen 3. Revolution Pi RevPi Core 3.	18
Imagen 4. Dispositivo de entradas y salidas digitales RevPi DIO.	19
Imagen 5. Célula de carga empleada en el proyecto.	19
Imagen 6. Caja de conexiones.	20
Imagen 7. Convertidor USB-RS485.	21
Imagen 8. Cámara empleada para la obtención de la matrícula.	21
Imagen 9. Ejemplo de funcionamiento de la cámara con el sistema ANPR que posee.	22
Imagen 10. Barrera automática PARK 30 XT PLUS.	23
Imagen 11. Detector de lazo inductivo empleado en el proyecto.	24
Imagen 12. Imagen del sensor fotoeléctrico empleado en el proyecto.	24
Imagen 13. Tablet empleada en el proyecto.	25
Imagen 14. Árbol básico de un proyecto para Raspberry Pi.	27
Imagen 15. Árbol del proyecto realizado.	28
Imagen 16. Captura de la pantalla de trabajo con Node-RED.	29
Imagen 17. Ejemplo del empleo del módulo de mysql.	29
Imagen 18. Propiedades del módulo mysql.	30
Imagen 19. Captura de todos los módulos que posee el paquete node-red-contrib-modbus. .	31
Imagen 20. Ejemplo del empleo de módulos del paquete node-red-contrib-modbus que leen y escriben los datos.	31
Imagen 21. Ejemplo de pantalla de trabajo de MySQL Workbench.	32
Imagen 22. Imagen de un camión rígido con volquete.	33
Imagen 23. Cabeza tractora con semirremolque.	33
Imagen 24. Camión descargando en un muelle de descarga.	34
Imagen 25. Descarga de camión en una zona de descarga.	34
Imagen 26. Programa principal del proyecto.	38
Imagen 27. Etapa de registro con los dos programas.	39
Imagen 28. Etapa de apertura de la barrera.	40
Imagen 29. Etapa Compara del programa.	41
Imagen 30. Etapa de comparación de los datos obtenidos al comparar.	41
Imagen 31. Propiedades para conectar los servidores de CODESYS con Node-red.	43
Imagen 32. Programación de un nodo para que lea una variable.	44
Imagen 33. Configuración de un nodo de salida de Node-RED.	45
Imagen 34. Proceso para guardar el dato de la variable matricula en Node-RED.	46
Imagen 35. Función para registrar datos en la base de datos.	46
Imagen 36. Propiedades inscritas para la conectividad del programa con la base de datos.	47
Imagen 37. Función para comprobar que los datos se han registrado correctamente.	47
Imagen 38. Programación del nodo que envía la señal de 1 al nodo de comunicación con CODESYS.	48
Imagen 39. Función para obtener el ID con los datos registrados.	48
Imagen 40. Función para obtener el Id de los datos obtenidos de la base de datos.	48
Imagen 41. Comandos para obtener el peso con respecto al ID.	49
Imagen 42. Comandos para obtener el peso con respecto al ID y la matrícula.	49
Imagen 43. Comandos para obtener el peso con respecto al ID y el peso.	50

Imagen 44. Función para mostrar el peso desde la base de datos.....	50
Imagen 45. Programación del nodo que envía cero a los nodos de comunicación.....	51
Imagen 46. Creación de la base de datos.....	52
Imagen 47. Creación de un nuevo usuario para trabajar con la base de datos desde otra IP. ..	52
Imagen 48. Privilegios otorgados al usuario para trabajar con la base de datos.	53

Índice de tablas del proyecto.

Tabla 1. Datos para programar las variables de entrada en Node-RED.....	44
Tabla 2. Datos para programar las variables de salida en Node-RED.	45
Tabla 3. Tabla de los elementos empleados en el proyecto y su coste.	121
Tabla 4. Tabla de representación del tiempo empleado y su coste.....	122
Tabla 5. Tabla que muestra el coste final del proyecto.	122



Memoria



Índice memoria.

1. Resumen.	15
2. Justificación.	15
3. Objetivo.	16
4. Alcance.	17
4.1. Hardware del proyecto.	17
4.2. Software del proyecto.	25
5. Antecedentes.	33
6. Normativa.	35
6.1. Normativa redacción del proyecto.	35
6.2. Normativa programación del proyecto.	35
6.3. Normativa peso máximo autorizado para camiones.	35
6.4. Normativa cámara de seguridad.	35
7. Requisitos de diseño.	36
8. Problemas surgidos y soluciones.	36
9. Programa implementado.	37
10. Planificación.	53
11. Resumen de presupuesto.	54
12. Conclusión.	55
13. Bibliografía.	55

1. Resumen.

La función principal del presente proyecto es automatizar la recepción de materias primas en la industria. Automatizando la toma de datos y validación de los mismos de forma remota.

Para ello se ha creado un programa de automatización para un ordenador de placa reducida que actuará como controlador de todo el proceso, además se han seleccionado todos los periféricos que se deben de emplear para poder implementar el proyecto.

El programa realizado se ha creado empleando diferentes lenguajes de programación para cada una de las partes del programa del autómat, además se han empleado diferentes programas software permitiendo la comunicación entre ellos que permiten un abaratamiento de costes del proyecto.

Po último, en este proyecto se han descrito las condiciones necesarias que deben cumplir todas las partes implicadas en el proyecto y una aproximación del presupuesto con todos los elementos necesarios para poder llevar este proyecto a cabo.

2. Justificación.

La idea general de la que surge este proyecto es la de crear una red de comunicación rápida entre una empresa, los suministradores y el servicio de transporte para poder minimizar el tiempo perdido en la producción debido a la falta de stock. Es decir, cuando a la empresa le hace falta cierto material solo deberá indicarlo y de una forma automatizada se avisará y a todas las partes involucradas en dicho producto. En la imagen 1 se puede observar un diagrama de Venn de tres conjuntos para mostrar de una forma visual lo que se quiere crear.

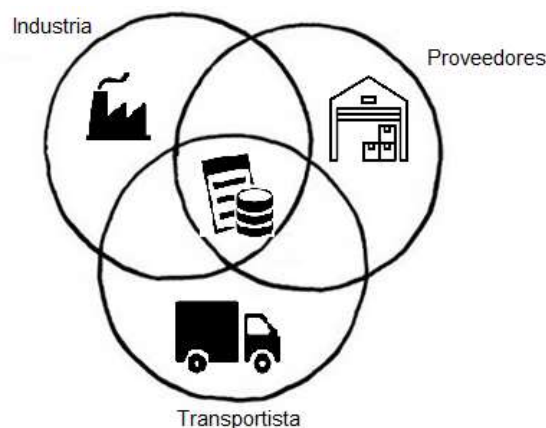


Imagen 1. Diagrama de Venn de 3 conjuntos que representa la idea general del proyecto.

Dentro de esta idea global surge la idea de este proyecto en la cual se quiere agilizar el acceso de los vehículos de transporte de mercancías a la empresa ya que se trata de una zona en la que suele haber mucha afluencia de tráfico, sobre todo en las grandes industrias, y por lo que

conlleva un tiempo de espera que se podría reducir mejorando el proceso de acceso y validación en el suministro de materias primas.

Para esta idea de proyecto se considera importante definir los datos que se van a obtener previamente para agilizar este proceso y como se obtendrían. Estos datos serían la matrícula del vehículo, el cual se obtendrá mediante la captura de la placa y su posterior transformación de la imagen en una serie de caracteres de forma automática mediante el sistema de reconocimiento automático de placas que posee la cámara, el peso obtenido mediante el pesaje del camión en una báscula y por último la identificación del conductor y del pedido de forma manual mediante un teclado.

Una vez obtenido estos datos serán enviados a un controlador, en este caso se emplea un Revolution Pi, que se trata de un ordenador de placa reducida. Este dispositivo tendrá el programa instalado y realizará las comunicaciones entre el programa y la base de datos donde registra o compara los datos.

Este programa se realizará mediante el software de automatización CODESYS, mientras que para la base de datos se emplea los programas de MySQL, con los cuales se puede crear y trabajar con bases de datos relacionadas entre sí y permite la comunicación con otros programas. En este proyecto se emplea para dicha comunicación el lenguaje de programación denominado Node-RED que permite recibir y enviar datos desde CODESYS a MySQL y viceversa, tal como se puede ver en la imagen 2.

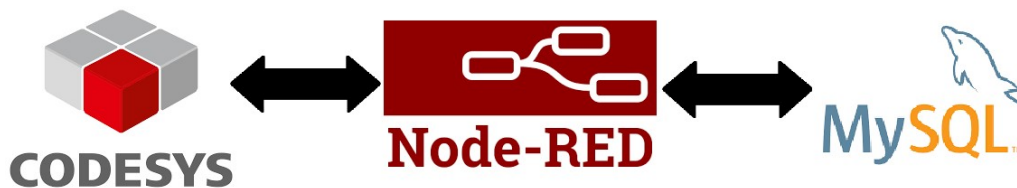


Imagen 2. Representación visual de la función que realiza Node-RED en este proyecto.

3. Objetivo.

El objetivo principal del proyecto es el desarrollo de un programa de integración de sistemas para plataformas de interproveedores. Así pues, este proyecto requiere obtener datos de los proveedores y de los transportistas para su validación en la entrega. Para ello se realizará un programa completo para el ordenador de placa reducida (SBC) que sea capaz de realizar todas las tareas que se le programen necesarias en el proyecto.

Cabe destacar la importancia en este proyecto de la comunicación entre los diferentes periféricos empleados mediante diferentes sistemas de comunicación, no solo empleando señales analógicas, sino también comunicación con conexiones RS485 o mediante conexión Ethernet, por lo que este proyecto se engloba dentro del Internet de las Cosas (IoT) o más concretamente del Internet Industrial de las Cosas (IIoT).

Para ello es necesario el intercambio de datos con plataformas a través de la red, ya que en dicho proyecto requiere comunicación con la base de datos vía red, la comparación de dichos datos y también el envío de señales desde la red al SBC.

También es objetivo de este proyecto la selección de los elementos necesarios para su implementación y que deben de cumplir con todas las especificaciones del mismo.

Por tanto, es necesario aplicar todos los conocimientos necesarios relacionados en el campo de la automatización industrial. Así mismo se trata de poder afianzar dichos conocimientos y ampliarlos dentro del mundo laboral desarrollando un proyecto que pueda solucionar problemas que pueden aparecer en las empresas.

4. Alcance.

En este apartado se van a describir por un lado los diferentes elementos empleados para poder llevar a cabo el proyecto y, por otro lado, también se describen los programas que conforman el software de este proyecto, es decir, se describirán los programas que se han empleado para la creación del programa del autómat.

4.1. Hardware del proyecto.

Primeramente, se han descrito los diferentes elementos hardware que conforman el proyecto, es decir, los diferentes elementos físicos necesarios para poder proyectar este proyecto. Por lo tanto, se van a describir desde el ordenador de placa reducida que hace de controlador del proyecto, hasta los elementos que envían o reciben las señales a dicho controlador.

4.1.1. Revolution Pi.

Revolution Pi es una plataforma de mini PCs industriales modulares desarrollada por KUNBUS basados en la Raspberry Pi. La Raspberry Pi se trata de un ordenador de placa reducida (SBC) de bajo coste y cuyo sistema operativo es el Raspbian con unas modificaciones realizadas por KUNBUS que permiten trabajar en tiempo real. Por emplear el sistema operativo Raspbian, las Revolution Pi permiten instalar software empleados en Raspberry Pi y S.O. Linux / Debian.

A diferencia de la Raspberry Pi, la Revolution Pi está basada en esta placa adaptada para cumplir con la normativa EN 61131-2 y IEC 61131-2 relacionadas con la industria. Por ello, se emplean para el Internet de las cosas industrial, *Industrial Internet of Things* (IIoT), ya que permite el envío de datos desde el PLC a la nube, permitiendo tener dichos datos disponibles en cualquier momento y emplearlos si son necesarios.

Otro aspecto importante de estos SBC y que ya se ha comentado anteriormente es que se puede emplear para el Internet de las cosas industrial, la cual consiste en maquinaria conectada a Internet y en plataformas de análisis que permiten emplear los datos que se producen en otros aspectos. Estos dispositivos IIoT abarcan desde pequeños sensores hasta grandes infraestructuras industriales.

Dentro de la gamma de Revolution Pi surgen diferentes modelos de módulos base que se pueden emplear según la necesidad de la empresa a instalar. En este proyecto como se puede ver en la imagen 3 se ha empleado RevPi Core 3.



Imagen 3. Revolution Pi RevPi Core 3.

Este módulo base como todos los de la gamma Revolution Pi comparten las siguientes características: el sistema operativo empleado es el Raspbian adaptado con RT_Patch, procesador BCM2837 con 4 núcleos, frecuencia de reloj de 1.2 GHz, memoria RAM de 1 GByte y memoria Flash eMMC de 4 GByte. En cambio, los módulos base de la Revolution Pi solo se diferencia por los conectores externos que poseen cada uno, en este caso, los conectores que tiene son: 2 x USB 2.0 A (500 mA de carga máx. por puerto), 1 x Micro-USB, HDMI, Ethernet (RJ45) 10/100 Mbits/s.

Además de los módulos bases como el comentado anteriormente, existe también una gamma de módulos de comunicación y módulos de entradas y salidas tanto digitales como analógicas.

Los módulos de comunicación se utilizan para conectar la Revolution Pi a diferentes redes industriales. De estos módulos se pueden conectar hasta dos módulos por sistema.

En cuanto a los módulos de entrada y salida que encontramos dentro de la gama se pueden clasificar en módulos donde todas sus conexiones son entradas o salidas analógicas o digitales, pero también hay en los que hay tanto entradas como salidas analógicas o digitales, pero todas del mismo tipo.

Para este proyecto es necesario un módulo de E/S digitales, por lo que se emplea un RevPi DIO como el que se puede observar en la imagen 4.



Imagen 4. Dispositivo de entradas y salidas digitales RevPi DIO.

Este módulo, como se ve en el esquema de conexiones que aparece a continuación (imagen 3), posee 14 canales de entrada a una corriente de 2.4 mA cuando esta alimentado a 24 V y 14 canales de salida con una corriente de salida máxima de 500 mA si se encuentra en modo alto y de 100 mA si se ha seleccionado el otro método de salida.

A parte del módulo comentado anteriormente también se ha empleado el convertidor USB-RS485 de la marca KUNBUS que se explica posteriormente en el apartado de la báscula, ya que se emplea para conectar el transmisor de peso con la Revolution Pi.

Para poder trabajar tanto con el convertidor USB-RS485 como con el RevPi DIO hay que programar la Revolution Pi en el programa PiCtory al cual se accede mediante la IP de la Revolution vía web y solo hay que agregar el módulo que se vaya a emplear.

4.1.2. Báscula.

Otro elemento del hardware del proyecto es la báscula para poder realizar los pesajes de los camiones. Para ello se empleará 4 células de carga de compresión de acero inoxidable 17-4 PH con una tensión de alimentación máxima tolerada de 15 V de la marca LAUMAS como se puede observar en la imagen 5.



Imagen 5. Célula de carga empleada en el proyecto.

Como en este proyecto se está realizando para la entrada de camiones de 2 o más ejes pero sin remolque se ha consultado que el peso máximo autorizado según la orden PRA/499/2017, de 1 de junio, por la que se modifica el anexo IX de Reglamento General de Vehículos, aprobado por el Real Decreto 2822/1998 de 23 de diciembre no supera las 32 toneladas, por lo que para la instalación de estas bascula se ha empleado las celdas cuya capacidad de cada una será de 10000 kg.

Estas celdas se caracterizan por que el error que surge al combinarlas es inferior o no supera el $\pm 0.03\%$ del fondo de escala de los 10000 Kg. Además, también poseen una buena protección para trabajar en ambientes externos, ya que son resistentes al polvo y tienen una buena resistencia al agua.

Estas 4 células que conforman la báscula se conectan a una caja de conexión como la que se puede ver en la imagen 6.



Imagen 6. Caja de conexiones.

Esta caja de conexiones está formada por una caja de protección con 4 pasacables para las celdas y 2 pasacables para la alimentación de la placa de conexiones. Esta caja de conexiones es la CLM8 de la marca LAUMAS, la se caracteriza porque posee 8 canales independientes para las células de carga lo cual permite el uso de funciones avanzadas como la ecualización digital, análisis del reparto de la carga y diagnóstico automático. También cabe indicar que esta caja de conexión puede realizar pesajes en zonas inclinadas de ± 10 grados y posee una memoria capaz de almacenar los últimos 50 pesajes realizados.

Por otro lado, también se ha seleccionado este modelo porque posee un puerto serie de salida RS485/RS232 con diferentes velocidades de transmisión a elegir en bit/s para la comunicación a través de protocolos ModBUS RTU, lo que permitirá una conexión con la Revolution Pi de forma directa empleando el siguiente conversor USB que podemos ver en la imagen 7.



Imagen 7. Convertidor USB-RS485.

Con este conversor e instalando en la Revolution Pi y agregando el módulo virtual Modbus RTU Master en el PiCtory se podrá obtener el pesaje de la báscula por la conexión RS485 y no de forma analógica ya que la conexión RS485 posee un mayor aislamiento frente al ruido.

Otro tipo de salida que puede tener esta caja de conexiones es una salida tipo Ethernet la cual se instala en la caja bajo pedido y con la cual incluiría este dispositivo dentro de los elementos que conforman el Internet de las cosas.

Por último, otros aspectos importantes de este dispositivo es que cumple las normativas 2014/31/EU - EN45501:2015 – OIML R76:2006 sobre la comercialización de instrumentos de pesaje de funcionamiento no automático. También posee diferentes modos a funcionar de la caja de conexiones, aunque en este proyecto se emplearía la función de rango único y cuya precisión es III.

4.1.3. Cámara.

Otro de los elementos del hardware del proyecto es la cámara empleada para la captura de la matrícula de los camiones. En este proyecto se emplea una cámara de la marca HIKVISION modelo DS-2CD7A26G0/P-IZ(H)S como la que se observa en la imagen 8.



Imagen 8. Cámara empleada para la obtención de la matrícula.

La principal característica por la cual se ha seleccionado esta cámara es por que posee un sistema de reconocimiento automático de matrículas incorporado. Un sistema de reconocimiento automático de matrículas, *automatic number plate recognition* (ANPR), es un software basado en

el sistema de reconocimiento óptico de caracteres, *optical character recognition* (OCR), que permite leer y registrar los caracteres de las matrículas. Este sistema capta la imagen digital de la matrícula por medio de la cámara, posteriormente realiza la transformación a una cadena de caracteres con la que se puede trabajar enviándola vía a Ethernet donde se necesite, como en este proyecto se va a hacer, o en la misma cámara como se ve en la imagen 9.

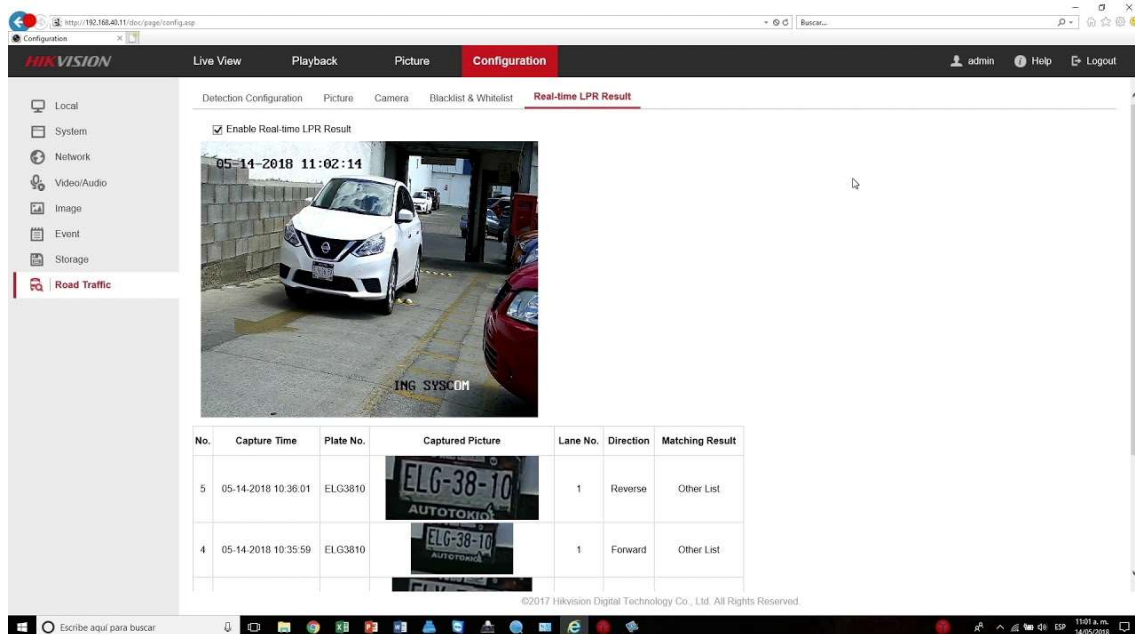


Imagen 9. Ejemplo de funcionamiento de la cámara con el sistema ANPR que posee.

Para este proyecto se emplea dicho software para la captura de las matrículas de camiones y también para realizar videograbaciones, por lo que se han tenido en cuenta el Reglamento (UE) 2016/679 el cual deroga el Reglamento General de Protección de Datos (RGPD), el cual indica todas las normativas que debe de cumplir una empresa para instalar un cámara de videovigilancia.

Otra característica a destacar de este tipo de cámara es que además de 2 entradas y 2 salidas digitales, también posee una conexión Ethernet por lo que esta cámara se puede comunicar dentro de los elementos que poseen el Internet de las cosas, el cual se explica en el apartado 3.1.1. del proyecto donde se describe la Revolution Pi.

A parte de las dos características más significativas que se han comentado anteriormente, hay que señalar que esta cámara tiene una resolución máxima de imagen de 1920x1080 y 50 fps a una frecuencia de 50 Hz. También hay que ver que esta cámara tiene una probabilidad de captar una imagen en parado del 98% y si el vehículo está en movimiento esta probabilidad se ve reducida al 96%.

En cuanto a las características de resistencia, posee una protección IP67, es decir, esta cámara posee una protección total contra el polvo y además puede ser sumergida en agua hasta profundidades de un metro. En conclusión, esta cámara posee una buena protección para trabajar en ambientes al aire libre. También posee una protección IK10, esta quiere decir que posee una buena protección contra golpes directos.

4.1.4. Barrera automática.

El último elemento importante para automatizar este proyecto es la barrera automática PARK 30 XT PLUS de la marca Aprimatic, como la que aparece en la imagen 10, que permitirá el paso de los camiones cuando se han comprobado que los datos ingresados son correctos.



Imagen 10. Barrera automática PARK 30 XT PLUS.

Se trata de una barrera de control de paso de vehículos diseñada para zonas que posean un tráfico muy intenso y por lo tanto debe de trabajar a una alta velocidad de maniobra. Por estas características, esta barrera es perfecta para zonas industriales.

Se trata de una barrera que posee un motor asíncrono trifásico de 230 Vac autoventilado que aporta una potencia de 370 W. La longitud del brazo es de 3 metros y cuyo tiempo de maniobra es aproximadamente de 0.7 segundos lo que hace que esta barrera puede realizar alrededor de 20000 ciclos al día. Posee también una protección IP 54 lo que hace que posea una buena protección frente a elementos externos.

Además de las características técnicas anteriores, también se destaca por el cuadro de maniobra con invertir para poder regular la velocidad del motor programable con funciones de declaración en los finales de recorrido y sistema de detección de obstáculos. Este cuadro de maniobra es la TRAFFIC VF.

Este cuadro de maniobra posee una lógica de control por microprocesador y señales luminosas para indicar en qué estado se encuentran las entradas, una pantalla con 3 dígitos para programar dicha placa e indicar el estado en que se encuentra el sistema. También posee 4 salidas configurables y un calefactor incorporado que permite poder trabajar con esta placa de control en zonas de baja temperatura. Este cuadro se alimenta a 230 Vac, igual que el motor de la barrera. Además, posee una salida a 230 Vac con una potencia máxima de 40 W y el resto de salidas son de 24 Vac y una intensidad máxima de 1 A.

A parte de los elementos comentados anteriormente que conforman la barrera automática, también cabe indicar tres elementos necesarios para controlar la zona de acceso y más específicos para la seguridad de esta zona.

El primer elemento a hablar es el lazo inductivo, el cual permitirá el cierre de la barrera automática. Para ello se ha empleado el detector LDD1PA2DU24 como el de la imagen 11 y un cable pelado de 1.5 mm² de diámetro. Este dispositivo detectará cuando el vehículo este encima del cable, el cual está enterrado en el pavimento, por medio del principio de inducción. Esto permitirá el cierre de la barrera cuando haya pasado el camión. La configuración de dicho detector y el proceso para instalar dicho detector y el cable se explica en el manual del detector que se puede ver en el anexo de los elementos del proyecto.



Imagen 11. Detector de lazo inductivo empleado en el proyecto.

Por otro lado, también se instalará un sensor fotoeléctrico polarizado modelo XUK9APANM12 de la marca Schneider Electric como el de la imagen 12, el cual se empleará en caso de si en el momento de cierre del mástil de la barrera el sensor fotoeléctrico se activa enviará una señal a la placa de la barrera para que por seguridad vuelva a abrir el mástil hasta que se desactive la señal.



Imagen 12. Imagen del sensor fotoeléctrico empleado en el proyecto.

Se trata de un sensor polarizado que se posee una tensión nominal de alimentación de 12-24 V en corriente continua. La distancia de detección nominal es de 6 metros, pero cabe recordar que

al tratarse de un detector réflex polarizado hace falta un reflector polarizado como el que se indica en la hoja de característica del detector, la cual se puede ver en los anexos de los elementos que conforman el proyecto, modelo XUZY50.

Por último, falta explicar el semáforo que se instalara para mostrar de una forma visual cuando está permitido el paso o no. Para ello hay que instalar un semáforo y una placa central de mando del semáforo.

4.1.5. Tablet.

El último dispositivo necesario para la puesta en marcha de este proyecto es una Tablet necesaria para poder introducir los datos del DNI del conductor y el ID del pedido en caso de que se haya de comparar algún pedido. Por lo tanto, para este proyecto se ha seleccionado una Tablet modelo PIPO X4 64GB Intel Cherry Trail T3 Z8350 Cuatro Núcleos como la de la imagen 13.



Imagen 13. Tablet empleada en el proyecto.

Se trata de una Tablet con una pantalla de 10.1 pulgadas con una resolución que proporciona un alto contraste y unos colores más vivos. Además, posee unas propiedades que permite que sean resistentes a golpes y con un IP65 que indica que este dispositivo es totalmente estanco al polvo y además está protegido contra chorros de agua en todas las direcciones.

Como se ha comentado anteriormente esta Tablet será para introducir los datos que debe de registrar el conductor. Para ello desde la Tablet se visualizará vía web la visualización que se podrá crear mediante el apartado de CODESYS que permite crear visualizaciones y trabajar con el proyecto. Para poder ver esta visualización se debe vincular la Tablet con una red Wi-Fi o se debe de insertar un adaptador para poder conectar dicho dispositivo mediante vía Ethernet al switch del proyecto.

4.2. Software del proyecto.

Ahora se va a describir todos los programas que conforman el software de este proyecto, es decir, los programas necesarios para que el controlador pueda realizar las tareas indicadas del

proyecto. En este caso se van a describir los programas de CODESYS y Node-RED con los que se va a realizar el programa implementado es este proyecto y los cuales trabajaran dentro del controlador empleado y también se explica el programa MySQL, con el que se crea la base de datos y con el que se realizaran comunicaciones.

4.2.1. CODESYS.

El programa principal empleado para la programación de este proyecto ha sido el CODESYS, ya que la mayoría de los programas de este proyecto se han escrito en esta plataforma de programación.

Se trata de una plataforma de programación empleada para los elementos de automatización industrial basada en la norma de programación IEC 61131-3, la cual se trata de una normativa que indica como programar los controladores lógicos programables de una forma lo más eficiente posible, y la cual permite una gran variedad de soluciones en proyectos de aplicaciones de automatización. Por lo tanto, se trata de un programa cuyo objetivo principal es el de proporcionar una herramienta para la implementación de tareas de automatización.

Se trata de una plataforma ideal para la Industria 4.0, la cual consiste en la fabricación informatizada con todos sus procesos interconectados entre si empleando el internet de las cosas, debido a las interfaces abiertas y los elementos de seguridad que llevan integrados. También permite trabajar con elementos que emplean la ingeniería de vanguardia, la comunicación entre elementos que se encentra en la red o nube y el intercambio de datos en redes IIoT.

Otra de las particulares de CODESYS es su gran flexibilidad para realizar los programas, ya que permite la programación en 5 lenguajes diferentes de programación, desde el lenguaje Ladder hasta el texto estructurado. Y también permite dentro de un mismo proyecto poder trabajar cada parte del proyecto en el lenguaje que mejor conlleve.

Por último, cabe indicar que se ha seleccionado esta plataforma de programación aparte de por las características mencionadas anteriormente, es por la gran variedad de PLC's que son compatibles con este programa ya que las grandes marcas confían en el programa estándar para programar sus controladores.

Para este proyecto a parte de emplear el programa estándar de CODESYS se ha instalado el paquete "CODESYS Control for Raspberry Pi SL" el cual permite poder trabajar con una Raspberry Pi como controlador. En este proyecto el controlador es una Revolution Pi de la marca KUNBUS, pero como se ha comentado anteriormente estos minis PCs están basados en una Revolution Pi, por lo tanto, este paquete es compatible con este mini PC.

A parte de las características descritas anteriormente que permite este paquete para trabajar con Raspberry Pi como controlador, también posee la capacidad de poder utilizar los módulos de extensión y varios dispositivos con los interfaces de comunicación SPI o I²C entre otros como podemos ver en la imagen 14, donde se puede ver el árbol básico de un programa con el paquete de "CODESYS Control for Raspberry Pi SL".

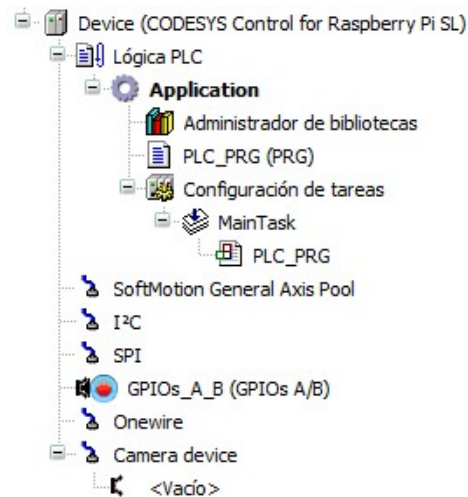


Imagen 14. Árbol básico de un proyecto para Raspberry Pi.

Otro elemento que hay que agregar al árbol de este programa es un dispositivo Ethernet el cual permite la comunicación entre diferentes dispositivos mediante este método de comunicación. Posteriormente también se agregare dentro del dispositivo Ethernet instalado un dispositivo modbus TCP esclavo, el cual se trata de un protocolo de comunicación estándar de factor en la industria basado en la arquitectura cliente/servidos (TCP/IP) y se caracteriza por su fácil implementación y su poco desarrollo, además de manejar bloques de datos sin suponer restricciones. Este dispositivo permite él envió de datos desde el programa realizado al Node-RED. Por lo que el árbol del programa de este proyecto quedaría de la forma que se observa en la imagen 15 y donde se podrían ver todos los dispositivos nuevos agregados, además del dispositivo de control se trata de una Raspberry Pi.

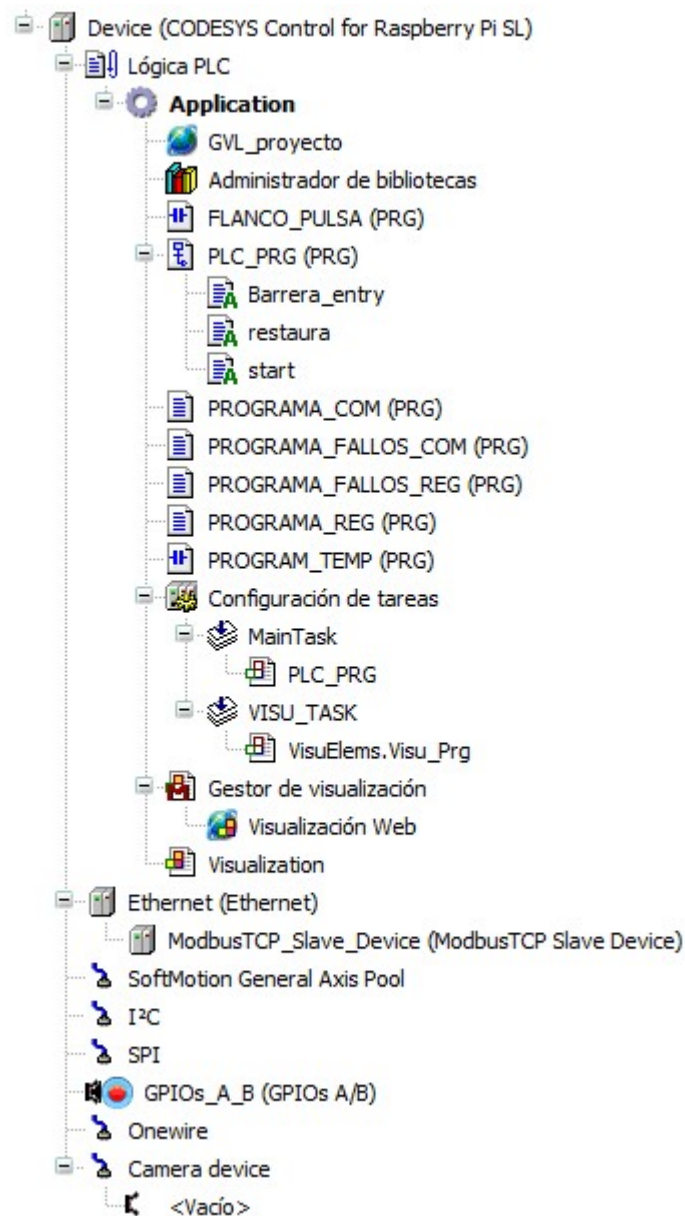


Imagen 15. Árbol del proyecto realizado.

4.2.2. Node-RED

Otro lenguaje de programación empleado en este proyecto es el Node-RED, el cual es una herramienta de programación que permite conectar dispositivos hardware, interfaz de programación de aplicaciones, “*Application Programming Interface*” (API), y servicios en línea. Node-RED es un lenguaje visual, que muestra las relaciones y funciones del programa realizado de una forma más dinámica, permitiendo al usuario la programación de una forma diferente a la programación de CODESYS, ya que en este programa no hace falta el empleo de ninguna de las lenguas especificadas en la norma IEC 61131-3.

Se trata de un programa editor de flujo que se visualiza por el navegador web, como se puede ver en la imagen 16, y cuyo funcionamiento consiste en la de ir añadiendo y eliminando nodos

que posteriormente se conectaran entre si con la finalidad de realizar la comunicación que se desea con dicho programa.

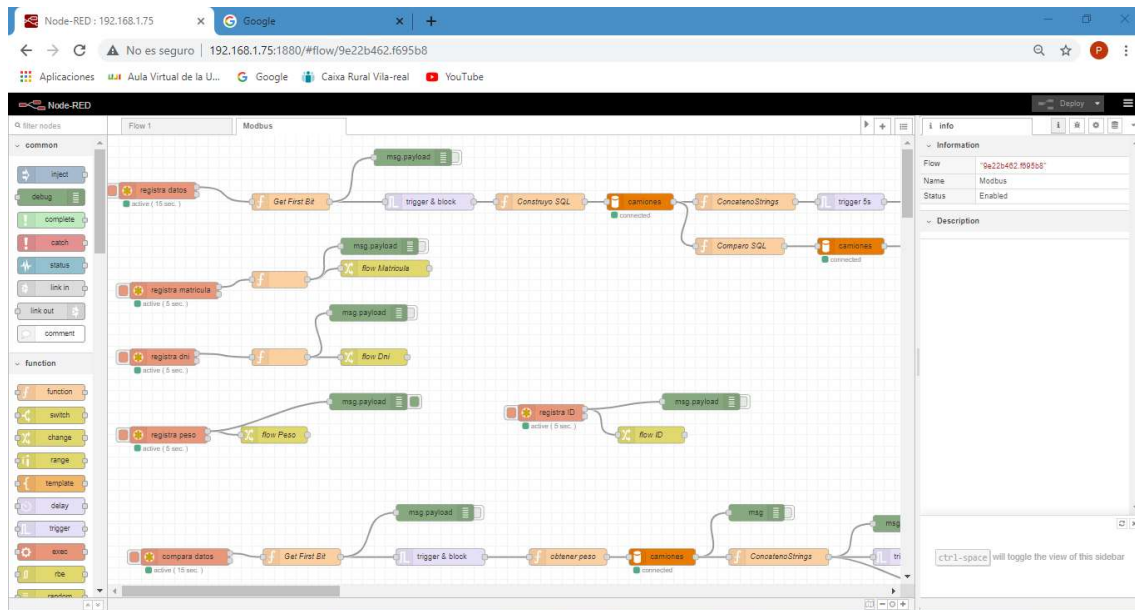


Imagen 16. Captura de la pantalla de trabajo con Node-RED.

El tiempo de ejecución se basa en Node.js, que aprovecha al máximo su modelo sin bloqueo controlado por eventos. Por tanto, hace este programa ideal para ejecutarse en todas las formas en la red, ya que puede ejecutarse en dispositivos de hardware de bajo coste como en una Raspberry Pi, así como en la nube.

Node-RED cuenta con más de 220000 módulos que se pueden instalar en forma de paquetes de Node y lo que permite ampliar la gama de nodos que se puedan utilizar en un programa y proporciona las mejores capacidades necesarias para el programa a crear.

En este proyecto, a parte de los paquetes básico que ya lleva instalado inicialmente se han instalado otros paquetes requeridos para la programación. El primer módulo instalado es el node-red-node-mysql, el cual permite la comunicación del programa con una base de datos que se halla creado con el programa MySQL. Para emplear este paquete solo hay que programar en el módulo la IP donde se encuentra la base de datos y el usuario y contraseña que permitirá el ingreso en dicha base de datos. En la imagen 17 se puede mostrar un ejemplo del empleo de este paquete en un proyecto en Node-RED, mientras que en la imagen 18 se puede ver las propiedades que se han inscrito para que el módulo funcione.

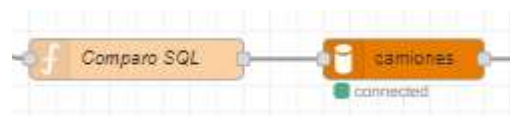


Imagen 17. Ejemplo del empleo del módulo de mysql.

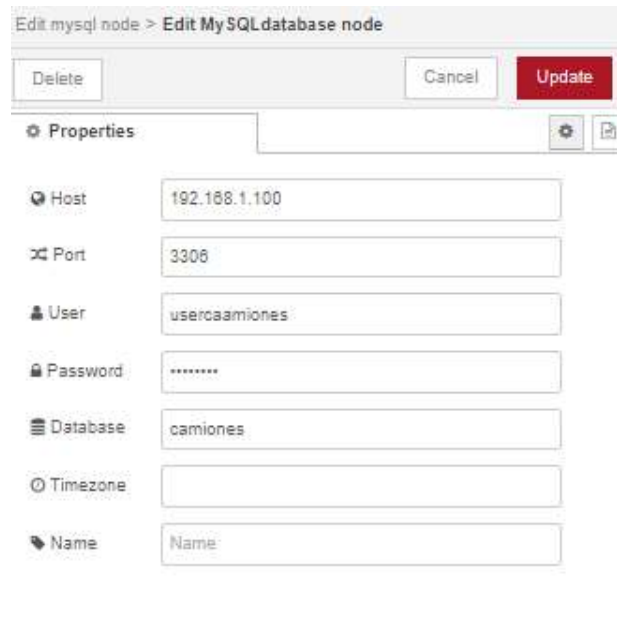


Imagen 18. Propiedades del módulo mysql.

El otro paquete instalado es el node-red-contrib-modbus el cual permite la comunicación entre el programa realizado por Node-RED y el programa realizado en CODESYS mediante la forma de comunicación TCP/IP, el cual consiste en la unión de dos protocolos diferentes.

El TCP, protocolo de control de transmisión, es un protocolo que se encarga de crear conexiones entre sí para que se pueda crear un flujo de datos. Esto garantiza que los datos se entreguen al destino sin errores y en el mismo orden que salieron.

El IP, protocolo de Internet, se encarga de enviar y recibir los datos en bloques, mientras que el envío lo realiza por la mejor ruta, pero no garantiza que todos los datos lleguen.

Por lo que el empleo conjunto de ambos protocolos permite que la información transmitida llegue siempre y empleando la mejor ruta.

En este paquete se encuentran tanto módulos que lean los valores de entrada y otros módulos que envíen los datos de salida como se pueden observar en la imagen 19.

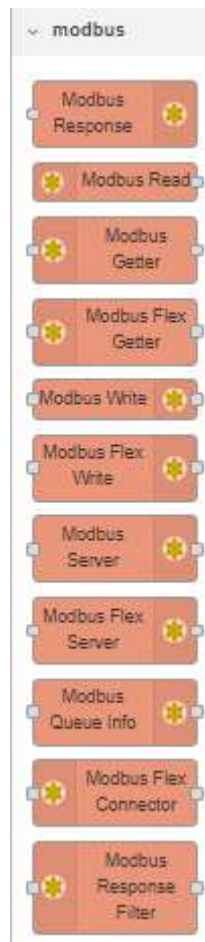


Imagen 19. Captura de todos los módulos que posee el paquete node-red-contrib-modbus.

Por otro lado, y como se observa en la imagen 20, el funcionamiento para el empleo de estos módulos es el mismo que al comentado anteriormente que emplea Node-RED en la cual se observa un ejemplo con un módulo que lee y otro módulo que envía los valores.



Imagen 20. Ejemplo del empleo de módulos del paquete node-red-contrib-modbus que leen y escriben los datos.

4.2.3. MySQL.

El último programa que se ha empleado en este proyecto es el MySQL el cual se trata de un sistema de gestión de bases de datos relacionales de código abierto, “*relational database management system*” (RDBMS). Se trata de un software para crear y administrar bases de datos con un modelo relacional utilizando diferentes herramientas como tablas, vistas, procedimientos almacenados y funciones entre otros como se puede ver en la imagen 21.

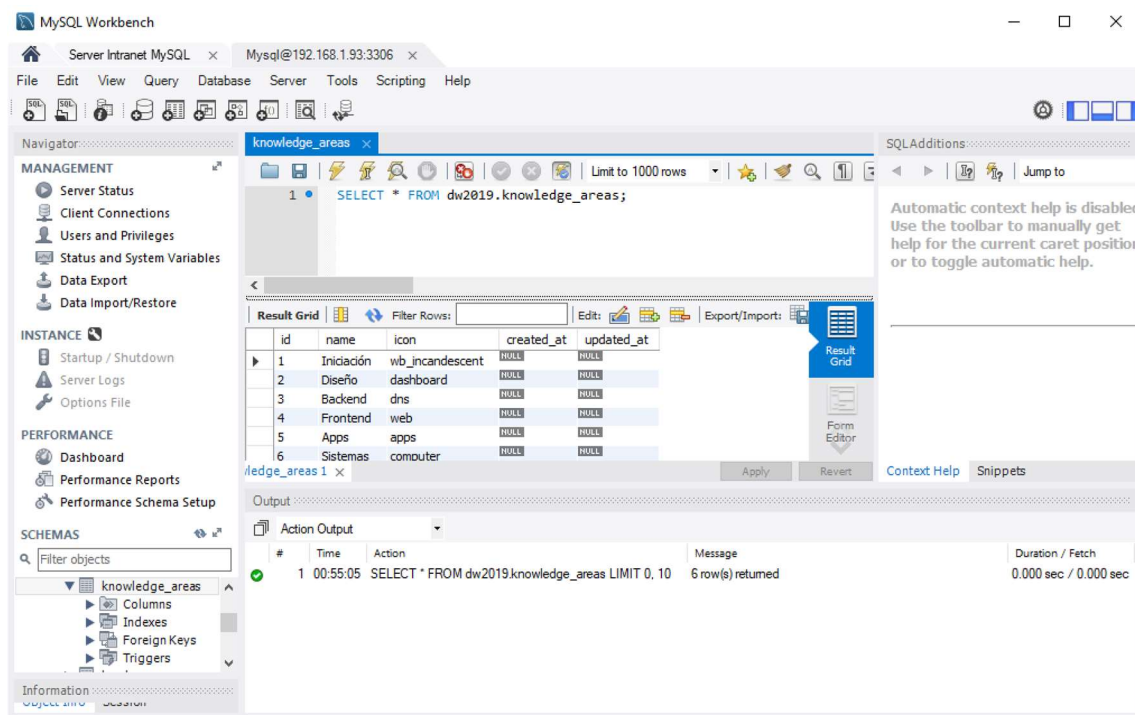


Imagen 21. Ejemplo de pantalla de trabajo de MySQL Workbench.

Un programa RDBMS está basado en el modelo cliente-servidor, es decir, el dispositivo u ordenador que tiene instalado y ejecuta el software MySQL es el cliente y siempre que quiera acceder a algún dato debe de conectarse con el servidor.

La comunicación de este software se realiza mediante un lenguaje específico del dominio denominado lenguaje de consulta estructurado, “structured query language” (SQL). Se trata de un lenguaje que se caracteriza por el uso del álgebra y el cálculo relacional para realizar consultas con la finalidad de recuperar información de una base de datos o realizar cambios en ella.

MySQL permite a parte de trabajar de una forma más simple con el panel de comandos en lenguaje C y C++, se puede trabajar de una forma más visual mediante el programa MySQL Workbench.

Una vez especificado algunos aspectos genéres importantes de MySQL cabe indicar que la finalidad por la cual se ha empleado este programa para la base de datos es porque:

- MySQL crea base de datos para almacenar y manipular datos, definiendo la relación de cada tabla.
- Los usuarios pueden realizar solicitudes escribiendo la instrucción SQL específica en MySQL y donde la aplicación responderá mostrándole la respuesta a su solicitud.
- Se puede realizar comunicaciones con el resto de programas empleados para realizar el programa tanto con CODESYS como con Node-RED.

5. Antecedentes.

Se pretende automatizar la entrada de los vehículos que transportan mercancías a las empresas para que de una forma rápida y sencilla se pueda agilizar esta tarea, ya que en la actualidad se trata de una de las partes más costosas de la empresa ya que desde el principio hay que diferenciar dos entradas diferentes, una para la entrada de camiones que transportan materia prima como puede ser arcilla o tierra y los cuales se transportan a granel o camiones que transportan materiales los cuales se miden en piezas o packs.



Imagen 22. Imagen de un camión rígido con volquete.

La primera entrada es para camiones rígidos con volquete, como el que se observa en la imagen 22, o cabeza tractora con semirremolque, como se observa en la imagen 23. Estos camiones transportan materiales que se compran a granel, es decir, estos materiales se compran al peso y por lo tanto hay un control de peso cuando llegan a las empresas.



Imagen 23. Cabeza tractora con semirremolque.

La entrada de estos vehículos a las empresas se realiza de la siguiente forma. Primeramente, el camión nada más llegar a las instalaciones es pesado para saber el peso total que llevan. Una vez pesado el camión se vacía y el transportista entrega los papeles del transporte al operario y este ya se encarga de transportar el material a la zona especificada para este.

Posteriormente el operario entregará todos los papeles de entrega a la oficina encargada del material, la cual se encargará de comprobar si efectivamente todos los datos que se hallan entregado en las hojas y el peso obtenido en la báscula se los correctos. También cabe indicar que, en el pesaje inicial del camión, el peso que se obtiene a la llegada de este a la fábrica es muy probable que el del camión no sea el mismo que cuando se ha realizado el primer pesaje, por lo que desde la empresa se indica un error por el cual si el peso es inferior a este se le reclamar a la empresa la cantidad que falte.



Imagen 24. Camión descargando en un muelle de descarga.

Por otra parte, y como se ha comentado anteriormente, también hay que indicar las entradas de material que llega a las empresas en piezas o en packs, las cuales cuando llega el camión se le indica al conductor para que estacione en un muelle de descarga, como se observa en la imagen 24, o en una zona de descarga como la de la imagen 25. Una vez llegue a esa zona los trabajadores se encargarán de descargar el material del camión.



Imagen 25. Descarga de camión en una zona de descarga.

Una vez descargado dicho camión se entrega por parte del conductor el albarán de entrega y se comprueba que la cantidad que se ha solicitado y la que se ha recibido. Si estos datos fueran erróneos se le comunicaría a la empresa que envía el pedido el fallo y habría que resolver el problema.

6. Normativa.

En este apartado todas las normas que se han de tener en cuenta tanto para la redacción del proyecto como para la programación y los elementos que se necesitan para realizarlo.

6.1. Normativa redacción del proyecto.

Norma UNE 157001:2014 “Criterios generales para la elaboración de proyectos”

Norma UNE 50132:1994 “Numeración de las divisiones y subdivisiones en los documentos escritos”

Norma UNE 50113-1:1992 “Documentación e información. Vocabulario. Parte 1. Conceptos fundamentales”

6.2. Normativa programación del proyecto.

Norma IEC 61131 Conjunto de normas y documentos técnicos desarrollados por la Comisión Electrotécnica Internacional (IEC) con la idea de estandarizar los autómatas programables.

6.3. Normativa peso máximo autorizado para camiones.

Orden PRA/499/2017 en la cual se modifica el anexo IX del Real Decreto 2822/1998 el cual se trata del Reglamento General de Vehículos.

6.4. Normativa cámara de seguridad.

Reglamento (UE) 2016/679 del parlamento europeo y del consejo de 27 de abril de 2016 relativo a la protección de las personas físicas en lo que respecta al tratamiento de datos personales y la libre circulación de estos.

7. Requisitos de diseño.

En este apartado se va a indicar como se han seleccionado las celdas encargadas que se instalaran en esta instalación además de las conexiones que se deben de realizar para poder comunicar las celdas con la placa de conexión y posteriormente esta con la Revolution Pi.

A parte de las indicaciones que se explicaran a continuación hay que indicar que a la instalación de esta báscula habría que diseñar el proceso de construcción de la báscula según las características de la empresa donde se instalara dicho proyecto. Esta parte del proyecto se debería de realizar viendo las condiciones donde se va a proyectar el proyecto y la idea que tiene el cliente para esta construcción.

La idea de este proyecto es diseñar una báscula para el pesaje de camiones de 2 o más ejes hasta un vehículo motor de 3 ejes con semirremolque de 2 o 3 ejes llevando, en transporte combinado, un contenedor o caja móvil cerrados, igual o superior a 20 pies y homologado para el transporte combinado, por lo tanto y siguiendo la orden PRA/499/2017 comentada en el apartado de normativas del proyecto y la cual fue redactada por el gobierno para indicar el peso máximo autorizado de estos vehículos (MMA) el peso máximo es de 44000 kilogramos.

Por lo tanto y observando el catálogo de las celdas que se adjunta en los anexos de los catalogo del proyecto se puede ver celdas de diferentes capacidades. En esto proyecto se han seleccionada cuatro celdas de una capacidad de 30000 kilogramos cada una.

Posteriormente se deben conectar las 4 celdas de carga a la placa de conexiones CLM8 explicada anteriormente y a la cual observando el dibujo de las conexiones en el manual de dicha placa se conectarán los 4 cables de cada una de las celdas y los cables de alimentación. Además de esta placa saldrán los cables que se conectarán con la Revolution Pi mediante la conexión RS485 al adaptador USB-RS485.

8. Problemas surgidos y soluciones.

En este apartado se va a explicar los problemas más importantes surgidos durante la realización de este proyecto y todas las soluciones que se han obtenido para poder realizarlo correctamente. Además, también se explicará alguna mejora que se puede realizar a este proyecto para que fuera más completo.

El primer problema importante que surgió fue a la hora de realizar el programa es con la biblioteca de comandos que posee CODESYS para poder comunicarse con MySQL. Se trata de una librería de pago y desde la empresa se indicó que se buscara alguna otra manera de realizar esta comunicación ya que no querían un aumento del presupuesto habiendo soluciones gratuitas. Como solución a este problema se decidió el empleo del programa Node-RED y el cual se ha explicado anteriormente de que se trata y sus características principales.

Otro problema se trató en este proyecto fue la selección de la forma de comunicación entre las células de carga y el ordenador de placa reducida Revolution Pi. En un principio se optó por el uso de un transmisor de peso analógico. Se trata de un dispositivo el cual recibe el valor del peso obtenido desde las celdas y posteriormente lo envía al módulo de entradas analógicas para poder trabajar con este dato.

Cuando se intentaba realizar estas conexiones se observó que el transmisor de peso necesita una caja de conexiones donde se conectan las diferentes celdas empleadas. Esta caja de conexiones transmite el peso mediante una comunicación RS485. Al ver que se empleaba esta comunicación y que la Revolution Pi admite este tipo de conexiones se decidió en conectar la caja de conexiones directamente con el SBC mediante la conexión RS485, ya que se trata de una conexión mejor que la analógica ya que no le afecta el ruido.

Estos fueron los problemas más destacados que surgieron durante la realización del proyecto y las soluciones que se optaron para solucionarlos.

Por otro lado, en este apartado también se van a describir alguna mejora que se pueden realizar a este proyecto con un poco más de tiempo en la programación del programa y datos internos de la empresa que quiere comprar dicho proyecto.

La mejora que se le puede realizar en este proyecto consiste en poder vincular la base de datos del programa con la base de datos donde se encuentran las características de los transportistas que tienen la empresa de transporte para ver si poseen la documentación correcta para llevar el material de dicho pedido. Para realizar esta nueva parte del proyecto surgirá el problema de ver cómo está organizado cada base de datos de cada empresa de transportes para realizar el programa a esa medida o crear una nueva base de datos donde guardar esos datos igual para todas las empresas.

9. Programa implementado.

En este proyecto se han empleado 5 lenguajes diferentes de programación, los cuales son:

- i. El *“sequential function chart”* (SFC), se trata de un lenguaje de programación gráfico.
- ii. Texto estructurado, *“structured text”* (ST). Se trata de un lenguaje de programación de alto nivel muy parecido a los lenguajes PASCAL o C empleando expresiones e instrucciones que permiten la programación de bucles.
- iii. el lenguaje Ladder (LD), que es un lenguaje gráfico basado en los esquemas eléctricos de control clásicos.
- iv. Lenguaje gráfico visual Node-RED.
- v. Lenguaje C y C++ el cual se emplea para las instrucciones para comunicarse con la base de datos MySQL.

El programa principal de este proyecto está realizado con el lenguaje SFC como se puede ver en la imagen 26. Ya que al tratarse de un lenguaje gráfico se puede ver claramente todas las etapas y transiciones en las que se divide este proyecto. Este programa se puede observar con más claridad en el punto 1.3. PLC_PRG en los anexos de programa CODESYS en la página 65.

En la etapa inicial se encuentra la acción de entrada de nombre start, la cual se puede leer en el punto 1.3.2 de los anexos del programa CODESYS (página 66). Se trata de una acción de entrada, la cual consiste en un programa que solo se repetirá una sola vez al activarse la etapa en la que se encuentra la acción. Se trata de un programa escrito en el lenguaje de programación ST y cuya función es borrar lo que se encuentre dentro de las variables principales con las que trabaja el programa y además tomaran valor FALSE el resto de variables booleanas que se pueden encontrar activadas en el momento de activarse esta etapa.

Una vez descrito la etapa inicial del programa principal, este se divide en tres ramas diferentes. La primera rama tiene la función de abrir la barrera de forma manual. La segunda rama en cambio, ara la comunicación con la base de datos en caso si lo que se pretende realizar es el registro de un nuevo envío. La última rama del proyecto realizara la comparación de los datos que recibe desde los dispositivos de entrada con los que se encuentran en la base de datos. Estas tres ramas se describirán más detalladamente a continuación.

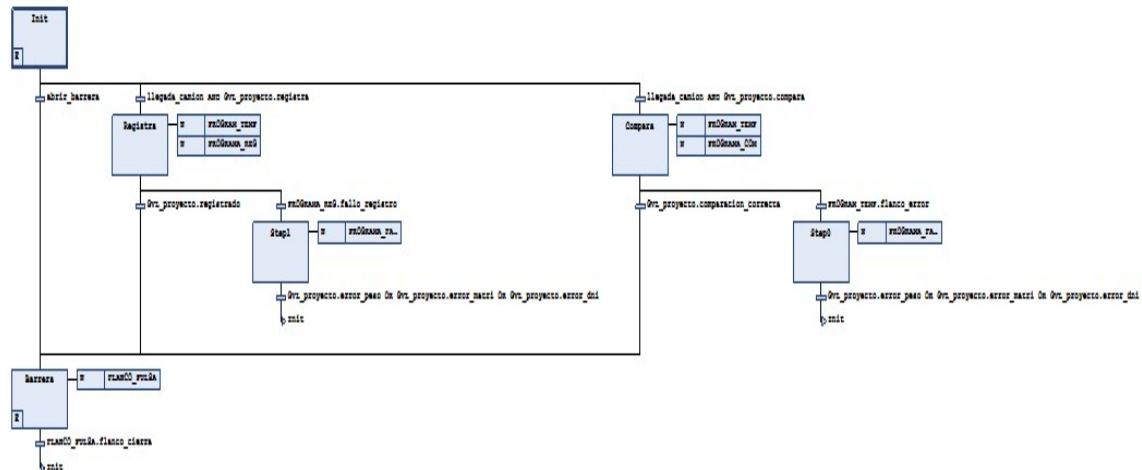


Imagen 26. Programa principal del proyecto.

La primera rama del programa como ya se ha dicho anteriormente sirve para si se quiere abrir de forma urgente o por si se produce un fallo de conexión con la red y no se pueda realizar la función deseada abrir la barrera automática con solo pulsa un pulsador, es decir, si se pulsa el interruptor abrir barrera se activará la etapa barrera, la cual se explicará posteriormente.

Para el resto de ramas del proyecto en la etapa inicial se obtendrán los datos necesarios de la siguiente manera. Cuando llega el camión y un sensor de tipo barrera lo detectara. Al mismo tiempo la cámara detectara por si sola la matrícula y realizara la transformación de la imagen a caracteres y la enviara al SBC vía Ethernet. Por otro lado, al detectar el camión encima de la báscula, el transmisor de peso enviará el valor obtenido al SBC mediante la conexión RS485. Por último, el conductor introducirá su DNI para poder identificarse.

Con todos estos datos el conductor pulsara el botón de registrar y activara la rama que permite registrar los datos de un nuevo envío que sale desde una empresa proveedora a la empresa que desea dicho producto. Para realizar esta tarea se activa la etapa de registro y como se observa en la imagen 27 se encuentran dos acciones que se tratan de dos programas, los cuales están relacionados entre sí.



Imagen 27. Etapa de registro con los dos programas.

El programa principal de esta etapa es el PROGRAMA_REG. El programa está realizado en el lenguaje de texto estructurado y se puede observar en el punto 1.7 de los anexos del programa CODESYS (página 71).

La primera tarea que realiza este programa una vez se activa es copiar los datos registrados en la etapa anterior del programa a las variables que se encuentran comunicándose en el programa Node-RED, mediante el modbus agregado al árbol del proyecto como se ha explicado en la parte de la memoria donde se habla del software CODESYS.

Este programa también lee las variables que se han escrito y si en alguna de estas no se ha registrado nada o 0 en la variable correspondiente al peso, esto querrá decir que hay un fallo de lectura de datos por lo que se activará la señal de error.

En cambio, si los datos leídos son buenos se activará la variable GVL_proyecto.registra para que en el programa PROGRAM_TEMP, punto 1.8 de los anexos del programa CODESYS (página 72), active un temporizador que pasado cierto tiempo activará la señal que se comunica con Node-RED del mismo modo que las variables anteriores que comunican con Node-RED.

Este temporizador es empleado para que Node-RED pueda realizar primeramente una lectura de las variables que tienen los datos de registros y posteriormente ya lea la señal para registrar los datos. Si no se hubiera realizado este paso Node-RED podría haber realizado un registro en blanco y que puede surgir que los datos no se hayan guardado en Node-RED.

Si estos registros han sido correctos, se enviará desde Node-RED una señal que se registrará en la variable registra_correcto descrita en el programa PROGRAMA_REG en CODESYS, además también se envía el ID del pedido en el que se halla registrado los datos introducidos. Este valor será mostrado por la pantalla. En este mismo momento, también se activará la señal que permitirá abrir la transición para llegar a la siguiente etapa del programa y una señal visual indicando que los datos se han registrado correctamente.

Si se activa la variable PROGRAMA_REG.fallo_registro indicará que hay un error y hará que se active una nueva etapa con una nueva acción asociada denominada PROGRAMA_FALLOS_REG descrito en el punto 1.6 de los anexos del programa CODESYS (página 70).

Este programa permite realizar comparaciones para ver cuál es el elemento que no se ha registrado, es decir, buscare cual es aquella variable que este vacía, lo que hará activar la señal visual que permite ver que es el error. Esta señal se mantendrá activa hasta que vuelva a hacer un nuevo registro o una nueva comparación, ya que tanto en el programa PROGRAMA_REG como en el PROGRAMA_COM pondrá a FALSE todas las señales de error.

En cambio, si los datos se han registrado correctamente en la base de datos se activará la transición GVL_proyecto.registrado la cual permite la activación de una nueva etapa que se puede ver en la imagen 28 y se encarga de activar la barrera.

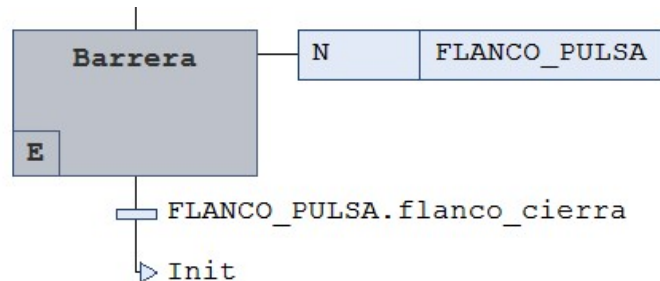


Imagen 28. Etapa de apertura de la barrera.

Cuando esta etapa se desactive querrá decir que la barrera se ha cerrado después de haberse abierto para el paso del camión. Cuando vuelva a la etapa inicial se mantendrá allí hasta que llegue un nuevo camión.

La última rama de este proyecto se ha diseñado para que se puedan comparar los datos introducidos con los que se encuentran en la base de datos de la siguiente forma.

Primeramente, cuando llega el camión se recibirán desde la cámara y del transmisor de peso los datos de la misma forma que para la rama de registro, al mismo tiempo el conductor introducirá su DNI y también el ID del pedido que está transportando. Cuando haya escrito todos los datos pulsara el botón de comparar, por lo que dentro del programa realizado se activara la etapa Compara.

En esta etapa, como se puede ver en la imagen 29, se activarán dos programas distintos pero que están relacionados entre sí. El programa PROGRAMA_COM es un programa realizado en el lenguaje de programación ST y que se puede ver en el punto 1.4 de los anexos del programa CODESYS (página 67). Se trata de un programa cuya función es enviar los datos a la base de datos para que sean comparados copiando los valores obtenidos en la etapa inicial a las variables que se encuentran en comunicación con Node-RED.

Al mismo tiempo en el programa PROGRAM_TEMP, el cual se trata de un lenguaje escrito en Ladder y la cual se describe en el punto 1.8 en los anexos de CODESYS página 72, y en cuya función es iniciar el contador TON_compara el cual se emplea con la función de esperar un cierto tiempo para indicar al programa que inicie la comparación para que se hayan podido registrar los datos de las variables en Node-RED. Este temporizador hace la misma función que el temporizador empleado para registrar datos en la base de datos.

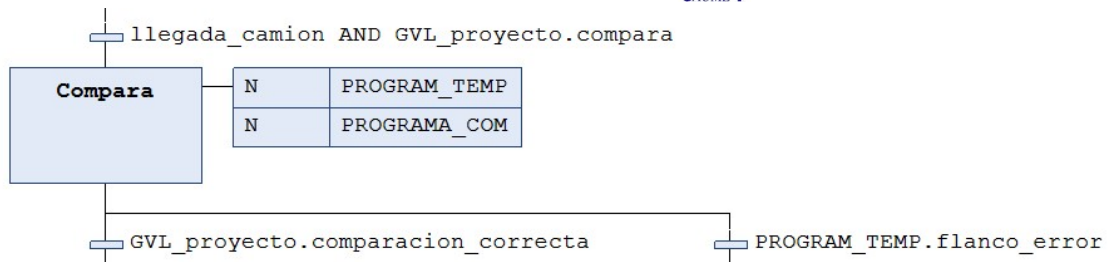


Imagen 29. Etapa Compara del programa.

Una vez se haya enviado la señal a Node-RED este realizará la tarea de comparar esos datos con los que se encuentran en la base de datos.

Posteriormente, este programa recibirá desde Node-RED los pesos de diferentes maneras para comprobar que los datos son correctos, la obtención de estos datos se explicará posteriormente cuando se hable del programa realizado con Node-RED.

Con todos los pesos obtenidos de la base de datos CODESYS comprobar que sean correctos. Primeramente, compara el peso obtenido solo con el ID con el recibido de la báscula, ya que es muy probable que ambos pesos sean distintos, y por lo que se ha introducido un error de $\pm 5\%$ para que si el peso obtenido en la báscula está dentro de este rango el peso será correcto.

A continuación, y si la comparación anterior es correcta se mira si todos los pesos obtenidos de la base de datos empleando la matrícula del camión y el DNI del conductor son los mismos. Si esta comparación es correcta se activará la señal de que los datos son correctos y por lo tanto se activará la etapa Barrera comentada anteriormente.

Si pasado un determinado tiempo, controlado por el temporizador que se encuentra en el programa PROGRAMA_TEMP, no se ha activado la señal para activar la etapa barrera esto querrá decir que hay algún dato erróneo porque no se habrán obtenido los pesos iguales. Por lo tanto, se activará la etapa que podemos ver en la imagen 30.

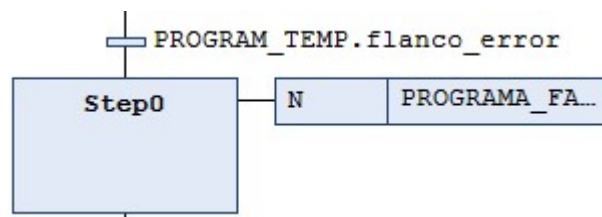


Imagen 30. Etapa de comparación de los datos obtenidos al comparar.

Se trata de una etapa que posee un programa asociado denominado PROGRAMA_FALLOS_COMP. Este programa escrito en el lenguaje de programación de texto estructurado, que se puede ver su estructura en el punto 1.5 de los anexos del programa CODESYS (página 69), es muy similar al programa de comparación de errores en el momento de registrar datos, pero en vez de ver si los datos están en blanco o está escrito el valor, aquí hay que distinguir los tres errores diferentes que se pueden encontrar.

El primer error es si el peso enviado por la báscula varía más del 5% del peso que se encuentra registrada en la base de datos. Ya que, si el peso es menor al indicado, esto querrá decir que habrá habido una gran variación de la cantidad de material comprado.

Por otra parte, el fallo de error del DNI o el fallo de matrícula sí que tienen el mismo funcionamiento, el cual consiste en comparar el peso obtenido de la base de datos empleando solo el ID con el obtenido mediante el ID y la matrícula o el DNI, ya que si empleando este segundo Node-RED no nos envía nada esto querrá decir que la matrícula o el DNI empleado no se corresponde con el registrado en la base de datos y por lo tanto este dato será erróneo. Si se activa alguno de estos errores se activará la señal luminosa que lo indicará y el programa volverá a la etapa inicial esperando nuevas indicaciones.

Por último, se va a comentar la última etapa de este programa, la cual se puede ver en la imagen 28 anterior. Se trata de una etapa que la emplean las tres ramas comentadas anteriormente y cuya función es la de abrir la barrera. Para realizar esta tarea se ha creado una acción de entrada asociado a esta etapa denominado `Barrera_entry`, que se describe en el punto 1.3.1. de los anexos del programa CODESYS en la página 66, escrito en texto estructurado. Esta acción, aparte de activar la señal de abrir la barrera se desactivarán todas las señales que estén activa de las etapas anteriores.

Aparte del programa comentado anteriormente denominado `Barrera_entry` también podemos observar en la imagen 28 que hay una acción asociada, la cual se trata del programa `FLANCO_PULSA`. Este programa, está escrito en el lenguaje de programación Ladder como se puede ver en los anexos del programa 1.2 en la página 64, tiene la función de detectar cuando hay un flanco de subida en la variable `PLC_PRG.barrera_cerrada`, que es la señal que recibe el SBC de la placa TRAFFIC VF de la barrera automática la cual está activa cuando la barrera está cerrada.

El programa `FLANCO_PULSA` comentado anteriormente se realizó porque la señal recibida desde la barrera suele estar activada ya que la barrera en condiciones normales estará cerrada, por este motivo si se llega a la etapa y esta señal está activada no se activará la etapa. Por este motivo, este programa espera a que la señal `PLC_PRG.barrera_cerrada` vuelva a activarse, mediante la detección del flanco de subida, para activar la variable de transición `FLANCO_PULSA.flanco_cierra`.

A parte de todos los programas ya nombrados anteriormente y cuya implementación se puede ver en el anexo del programa CODESYS, también hay un punto de este anexo el cual describe todas las variables globales que se emplean en este proyecto en el punto 1.1 de los anexos del programa CODESYS en la página 63, además podemos encontrar otras variables en algún programa específico ya que solo son empleadas en dicho programa.

Una vez ya explicado el programa principal realizado con CODESYS, se va a describir del programa realizado en Node-RED. Este lenguaje de programación, como ya se ha comentado anteriormente en la memoria del proyecto, consiste en diferentes bloques donde cada uno puede hacer una función que el programador puede programar y que se van relacionando entre sí aquellos que deben de tener una misma finalidad.

Primeramente, se van a describir los bloques encargados de recibir los datos desde CODESYS con el dispositivo Modbus TCP agregado a Ethernet en el árbol del programa principal. Hay que destacar que estos datos que se envían desde CODESYS son variables de diferentes tipos, pero

este Modbus solo envía variables tipo WORD, por lo que la conexión se realiza de diferente forma según el tipo de variable.

Las variables de matrícula y DNI son de tipo STRING, es decir, se trata de una cadena de caracteres que conserva un orden de asignación. En este programa el tamaño de estos STRING es de 16 bits, ya que en este tamaño cabe un dato de matrícula o DNI.

Por otro lado, las variables de ID y peso son datos UINT, mientras que las variables de comprar, registrar y reiniciar los datos son datos BOOL.

Antes de comenzar a escribir los datos para que reciba los datos hay que programar el nodo para que pueda comunicarse con CODESYS, para ello se programa el servidor como se ve en la imagen 31.

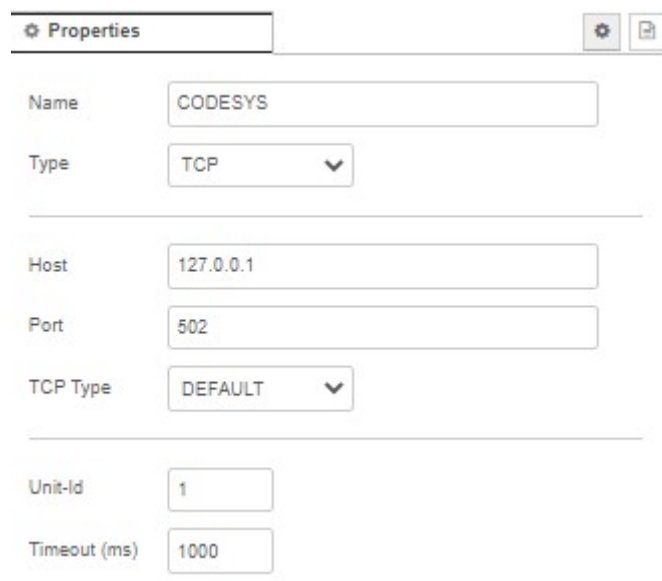


Imagen 31. Propiedades para conectar los servidores de CODESYS con Node-red.

Como se puede ver en la imagen los únicos datos que se deben de introducir son el IP en “host” donde se encuentra el programa de CODESYS, en este caso ambos programas se encuentran en la Revolution Pi, por lo que se puede escribir la IP local.

También hay que escribir el puerto en “port” y el “Unit-ID”, los cuales se han programado en el Modbus de CODESYS en el momento de su instalación.

Posteriormente a conectar ambos programas hay que indicar a cada nodo que deba de leer alguna variable de CODESYS donde se encuentra dicha variable. Estos datos se colocarán en la ventana que se puede ver en la imagen 32. Lo único que hay que hacer es indicar la función que debe hacer en FC, colocar la dirección de donde se encuentra el dato en el Modbus en “Address” y la cantidad de valores que tiene que leer en “Quantity”

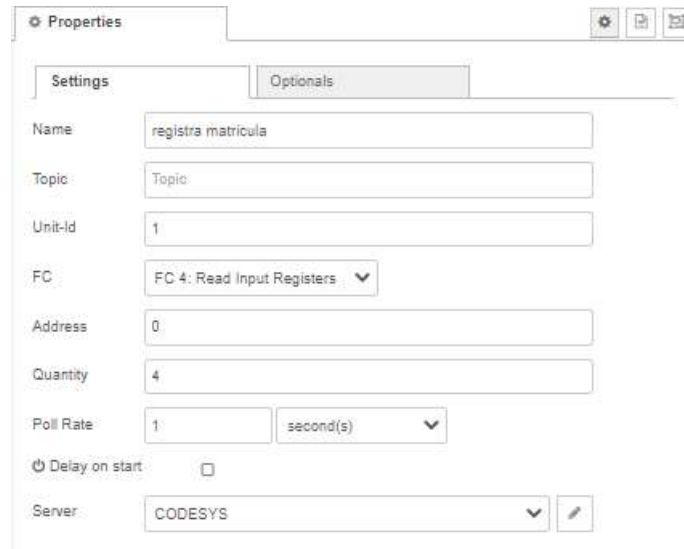


Imagen 32. Programación de un nodo para que lea una variable.

Como Node-RED debe de leer un total de 7 variables de CODESYS se ha creado la siguiente Tabla 1 para ver como se ha realizado la programación de cada una de estas variables. Por otra parte, en el punto 1 de los anexos del programa de CODESYS podemos ver como se han asignado cada variable a la dirección indicada para que se pueda realizar la comunicación entre el nodo de Node-RED y la variable en CODESYS.

Nombre variable	Tipo de variable (CODESYS)	Unit-Id	FC	Address	Quantity
Matrícula	STRING	1	4	0	4
DNI	STRING	1	4	4	5
Peso	UINT	1	4	9	1
ID	UINT	1	4	13	1
Registra datos	BOOL	1	4	11	1
Compara datos	BOOL	1	4	12	1
Reinicia datos	BOOL	1	4	14	1

Tabla 1. Datos para programar las variables de entrada en Node-RED.

Como se ve en la Tabla 1, los datos con variables tipo UINT o BOOL, el valor que se introduce en el apartado Quantity del modbus es 1, eso es debido a que el tamaño de estas variables es menor que el tamaño de una variable WORD. Por otro lado, las variables STRING hay que indicar cuantos WORD ocupa cada dato, ya que cada dos datos de un STRING equivalen a un WORD. Así pues, la variable matrícula con siete datos STRING les corresponden 4 variables WORD. Por otro lado, la variable DNI son 9 STRING por lo que corresponde a 5 variables WORD.

Por otro lado, también hay que programar los nodos que envían datos a CODESYS. En este caso los datos que recibirá el programa en CODESYS solo son variables BOOL o UINT, por lo tanto, en el espacio donde se escribe la cantidad de datos será siempre 1 por dicho motivo.

Los nodos de salida poseen una configuración muy similar a la de los nodos de entrada como se puede ver en la imagen 33. En esta configuración se debe de colocar el mismo servidor como el programado en la imagen 31, lo único que cambia en esto es la función que realiza y la dirección donde debe enviar cada uno de los datos al Modbus insertado en el programa principal. La configuración de todos estos nodos se puede ver en la Tabla 2, con la configuración empleada en este proyecto. La asignación de las direcciones en CODESYS se puede ver en los puntos 1.1 y 1.7 de los anexos del programa CODESYS (páginas 63 y 71).



Imagen 33. Configuración de un nodo de salida de Node-RED

Nombre variable	Tipo de variable (CODESYS)	Unit-Id	FC	Address
Datos correctos	BOOL	1	6	1
ID obtenido	UINT	1	6	3
Compara correctos	BOOL	1	6	2
Peso obtenido	UINT	1	6	4
Peso matricula	UINT	1	6	6
Peso DNI	UINT	1	6	5

Tabla 2. Datos para programar las variables de salida en Node-RED.

Una vez Node-RED puede leer los datos que recibe se crea una conexión de nodos para poder guardar los datos de la matrícula, DNI, peso e ID. Como se puede ver en el punto 2.1 de los anexos del programa Node-RED (página 73), la captura y memorización de las variables nombradas anteriormente se ha realizado de dos formas distintas. Para las variables de la matrícula y el DNI se observa que entre el nodo de leer los datos y el que los guarda se ha creado un nodo con una función que permite unir todas las variables WORD de una única variable con un único dato. Por otro lado, para las variables de peso e ID solo se ha unido el nodo que lee los datos con el que guarda, ya que no hace falta ninguna conversión.

En la imagen 34 muestra cómo se debería de configurar el nodo Flow matrícula que permite guardar la variable matrícula para poder trabajar con ella por todo el programa. El resto de nodos Flow del resto de variable se realizan de la misma forma, pero cambiando el nombre de matrícula por el nombre de cómo se quiera guardar la siguiente variable.

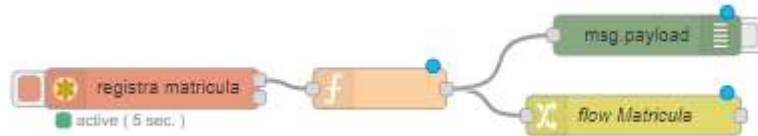


Imagen 34. Proceso para guardar el dato de la variable matricula en Node-RED.

Una vez se hayan creado todas las líneas para guardar los archivos ya se crean las líneas correspondientes para registrar los nuevos datos.

En esta primera, la cual se puede ver su programa en el punto 2.2 del anexo del programa de Node-RED (página 74), se colocará el nodo que lee el dato de cómo se encuentra la variable de registro en CODESYS. Posteriormente, se ha colocado un nodo de función que lo único que hace es leer el primer bit de la variable que recibes, ya que será el que indicará si se activará. Seguidamente se añadirá un nodo para que active todo lo que se encuentra después de él, una sola vez cada vez que se active el nodo registro.

Estos primeros pasos comentados anteriormente para la línea de comandos para registrar un dato es igual que para comparar o reiniciar los datos únicamente cambiando el nodo inicial con el correspondiente que posea las direcciones de memoria de la variable comprar o reinicia.

Después de los pasos se añade un nodo función donde se escribe los datos que se observan en la imagen 35 en la cual se llaman los datos guardados y se escribe la función como si estuvieras escribiendo en MySQL.

Name

ID MySQL

Function

```

1 var auxMatricula = flow.get("matricula");
2 var auxDni = flow.get("dni");
3 var auxPeso = flow.get("peso");
4 //var sql="SELECT mqtt.wago8202(mensaje) VALUES('"+msg.payload+"')";
5 var sql="SELECT id FROM camiones.entradas WHERE matricula='"+auxMatricula+"' AND dni='"+auxDni+"' AND peso='"+auxPeso+"'";
6 msg.topic=sql;
7 msg.matricula = auxMatricula;
8 msg.dni = auxDni;
9 return msg;

```

Imagen 35. Función para registrar datos en la base de datos.

Posteriormente a este nodo ya se coloca el nodo que permite comunicarse con la base de datos. En este nodo solo hay que indicar la dirección IP donde se encuentra la base de datos, el usuario y la contraseña para entrar y el nombre de la base de datos donde tiene que realizar dicho proceso como se puede ver en la imagen 36, donde se muestra las propiedades que se han registrado para el funcionamiento de la conectividad de Node-RED con MySQL.

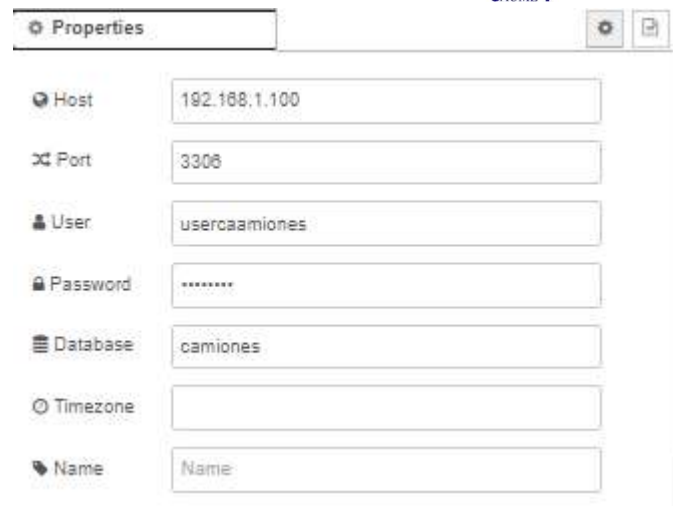


Imagen 36. Propiedades inscritas para la conectividad del programa con la base de datos.

Cuando ya se haya hecho el proceso anterior, esta línea de comandos se bifurcará en dos, una que enviara una señal a CODESYS que dirá que los datos se han registrado correctamente y otra línea que le pedirá a la base de datos que no indique el ID en el cual ha registrado dichos datos.

En la primera línea de datos se insertará un bloque de función el cual leerá una parte de la información obtenida de la base de datos cuya función se observa en la imagen 37. Posteriormente, se añadirá un nodo que cuya función consiste que cuando se haya obtenido la expresión que expresa la función anterior enviará una señal valor 1 que permanecerá activa durante 5 segundos y volverá a 0 al nodo de comunicación con CODESYS denominado datos correctos y cuyas especificaciones se encuentran en la tabla 2. La programación del nodo que envía 1 o 0 al nodo de escritura a CODESYS se puede ver en la imagen 38.



Imagen 37. Función para comprobar que los datos se han registrado correctamente.

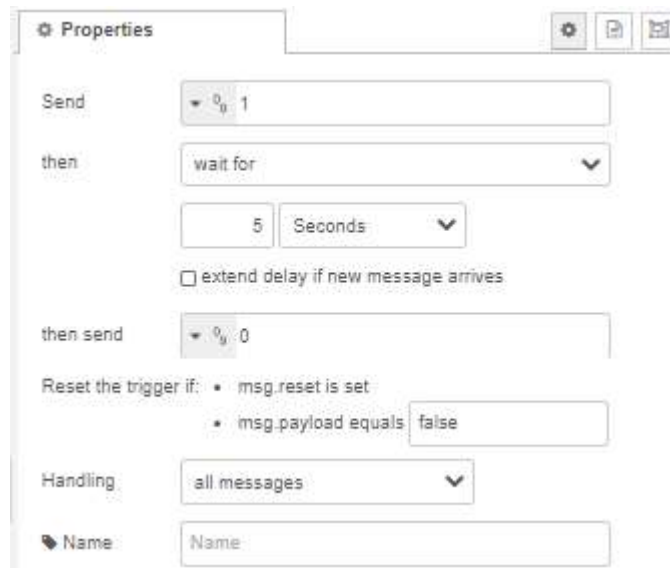


Imagen 38. Programación del nodo que envía la señal de 1 al nodo de comunicación con CODESYS.

En la segunda línea en cambio hay un bloque de función, el cual posee el programa que se ve en la imagen 39, el cual se obtendrá el ID de la nueva línea de datos ingresadas mediante el peso, la matrícula del vehículo y el DNI del conductor.

```
1 var auxMatricula = flow.get("matricula");
2 var auxDni = flow.get("dni");
3 var auxPeso = flow.get("peso");
4 //var sql="SELECT mqtt.wago8202(mensaje) VALUES('"+msg.payload+"');";
5 var sql="SELECT id FROM camiones.entradas WHERE matricula='"+auxMatricula+"'AND dni='"+auxDni+"' AND peso='"+auxPeso+"';";
6 msg.topic=sql;
7 msg.matricula = auxMatricula;
8 msg.dni = auxDni;
9 return msg;
```

Imagen 39. Función para obtener el ID con los datos registrados.

Una vez enviada la función de la imagen 39 por el nodo MySQL a la base de datos se obtendrá el resultado y mediante un nodo función se obtendrá dicho ID con la función que se lee en la imagen 40. Posteriormente, este valor será enviado a CODESYS mediante el nodo de comunicación id obtenido cuyas especificaciones se encuentran en la Tabla 2.



Imagen 40. Función para obtener el Id de los datos obtenidos de la base de datos.

Por otra parte, los comandos empleados para comparar los datos, los cuales se pueden ver en el punto 2.3 de los anexos del programa Node-RED (página 75), empiezan de la misma forma que para registrar datos como ya se ha comentado anteriormente. Una vez ahí, se ha bifurcado la línea de comandos en tres líneas diferentes ya que se van a crear 3 funciones para que obtengan el peso que está registrado en la base de datos de 3 formas distintas: la primera solo con el ID, la segunda con el ID y la matrícula, y por último con el ID y el DNI.

Se ha realizado la obtención del peso empleando estas funciones porque en el momento de enviar los datos a CODESYS donde realiza la comparación es más fácil el envío de una variable UINT, ya que el tamaño de estas variables es menor que el de una variable WORD y, por tanto, cabe dentro de esta variable y no obtener un STRING de Node-RED ya que tiene un tamaño de cinco variables tipo WORD. Por este motivo, se ha escrito tres funciones diferentes comentadas anteriormente y que a continuación se explica el motivo de cada una:

- La primera pide el peso solo con el ID introducido ya que este ID es único para cada pedido. La función empleada para este paso se puede ver en la imagen 41.
- La segunda función pide el peso en función del ID y de la matrícula por lo que si la matrícula introducida no corresponde a dicho ID la base de datos enviara un error a Node-RED. Esta función se puede ver cómo debe ser inscrita en la imagen 42.
- La tercera función pide el peso con el ID y el DNI, el cual si no corresponde al ID recibirá un error desde la base de datos igual como ocurre en la función correspondiente a la matrícula. Dicha función se puede ver en la imagen 43 de cómo debe de programarse en nodo de función de Node-RED.



Imagen 41. Comandos para obtener el peso con respecto al ID.

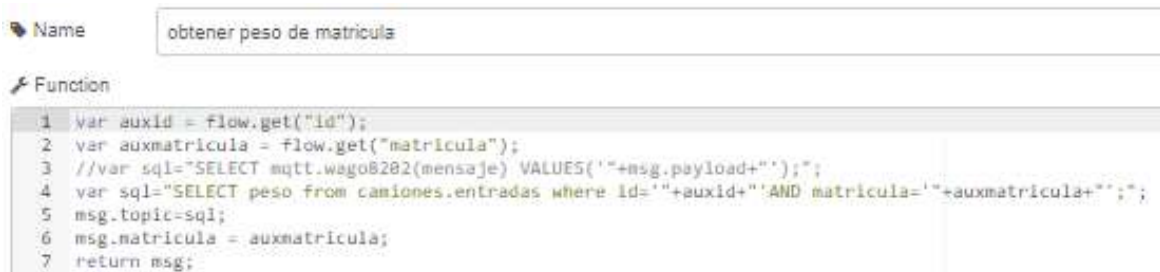


Imagen 42. Comandos para obtener el peso con respecto al ID y la matrícula.



Imagen 43. Comandos para obtener el peso con respecto al ID y el peso.

Una vez creadas las funciones se une con un nodo de conectividad con MySQL como ya se había realizado anteriormente para registrar los datos en la base de datos. A cada nodo de función se le debe de asignar un nodo para la base de datos ya que si no es así surge el error. Cada uno de estos nodos de MySQL posee las propiedades ya mostradas anteriormente en la imagen 36. Si ya se ha programado anteriormente algún nodo MySQL para que comunique con dicha base de datos estos datos ya estarán guardados.

Posteriormente a cada nodo de MySQL se añadirá un nodo función, el cual permitirá obtener el valor del peso que se ha pedido. Para que obtenga ese dato se deberá de inscribir en el cuadro del nodo a función los datos que aparecen en la imagen 44. Si dicho nodo no encuentra este comando será entonces cuando indique el error.



Imagen 44. Función para mostrar el peso desde la base de datos.

Para finalizar las ramas de comparación de datos solo hace falta relacionar las tres líneas creadas anteriormente con los nodos que envían estos datos al programa de CODESYS. Para ello, se relacionarán cada línea de comandos anterior con los nodos de comunicación explicados en la Tabla 2 de la siguiente forma:

- En la primera línea creada, la cual corresponde a la obtención del peso mediante el ID, se realizarán dos conexiones. La primera corresponde al nodo peso obtenido, mientras que la segunda corresponderá al nodo compara correctos. Antes de añadir este último nodo se añadirá un nodo que recibirá el dato del peso desde el nodo función comentado anteriormente y que enviará al nodo de comunicación con CODESYS un 1 durante 5 segundos y posteriormente volverá a 0. La programación de este nodo se encuentra detallada en la imagen 38.
- En la segunda rama, la cual corresponde a la obtención del peso mediante el ID y la matrícula, se añadirá el nodo de envío de datos peso matrícula.
- En la última rama correspondiente a la obtención del peso mediante el ID y el DNI del conductor, se colocará el último nodo que falta que se trata del nodo peso DNI el cual se ha explicado como el resto de estos nodos en la Tabla 2.

Por último, solo falta comentar del programa realizado con Node-RED la parte encargada de reiniciar los datos. Esta parte se encarga de volver a poner a cero los datos que se envían a Node-RED, ya que sino cuando se vuelve a activar el programa como ya encuentra unos datos guardados en las variables de CODESYS, no realiza correctamente la comparación. Este programa se encuentra detallado en el punto 2.3 de los anexos del programa Node-RED (página 75).

Para ello al bloque que recibe la señal para este proceso, el cual ya se ha descrito en la Tabla 1, se le añade un nodo función para que lea el primer bit recibido como en la rama de registrar o de comparar.

Posteriormente, se añade un nodo, el cual enviara un cero a los nodos de comunicación con CODESYS: peso obtenido, peso matricula y peso DNI. Para crear este nodo se puede observar su programación en la imagen 45.

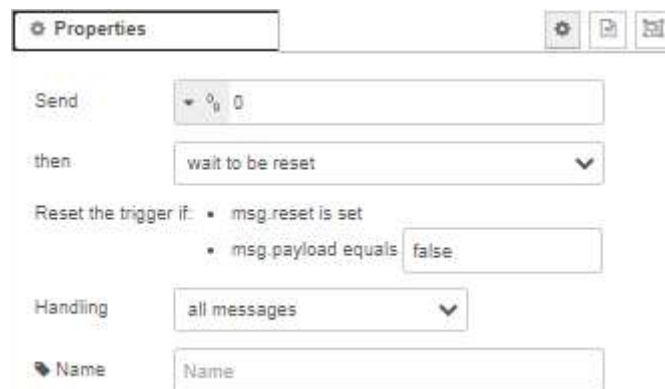


Imagen 45. Programación del nodo que envía cero a los nodos de comunicación.

Como se puede observar en todos los apartados de los anexos del programa Node-RED (páginas 73-75) aparecen unos nodos de color verde, los cuales no afectan en nada al programa, solo son nodos empleados para que muestren en una pantalla que aparece dentro de la aplicación empleados para crear el programa y ver si el funcionamiento es correcto.

Por último, para ya finalizar con el programa solo hace falta crear la base de datos mediante el programa MySQL Workbench que ya se ha comentado de que se trata dentro de la memoria.

Para este proyecto se ha creado una base de datos formada por 4 columnas, las cuales se van a describir una a una a continuación y se puede ver como se ha realizado su configuración en el programa en la imagen 46.

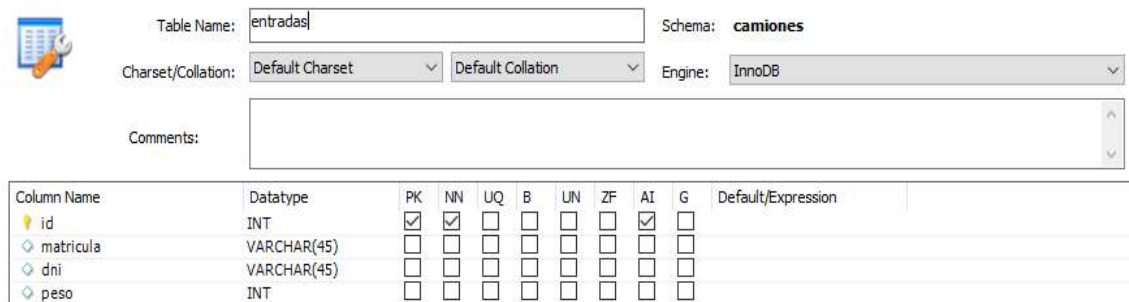
La primera columna de la base de datos corresponde al identificador del pedido, es decir el ID. Se trata de una columna de tipo entero, INT, que no se puede modificar, y que cuyo valor se va incrementando automáticamente cada vez que se añade una nueva fila.

La segunda columna corresponde a la matrícula del camión. Se trata de una columna tipo varchar (45), es decir una cadena de caracteres cuyo tamaño es de 45 bits. También hay que

indicar que se trata de una columna cuyos datos ingresados se pueden modificar o eliminar si es necesario.

La tercera columna corresponde al DNI del conductor. Es una columna igual que la de la matrícula del vehículo, es decir, se trata de una columna de la misma variable y las mismas características que la columna anterior.

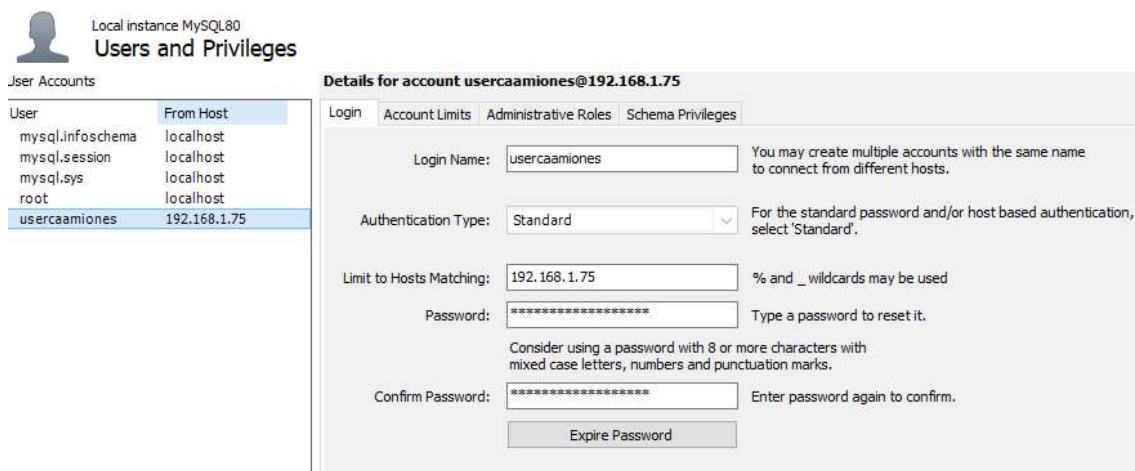
La última columna de esta base de datos es la que corresponde a la que registra el peso del camión. Se trata de una columna cuyos datos son tipo enteros, INT, y cuyas características en referente a lo que se puede realizar con los valores que se encuentran en esa columna son iguales que a la de ambas columnas anteriores.



Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
matricula	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
dni	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
peso	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Imagen 46. Creación de la base de datos.

Por último, también debe de crearse un usuario en el programa que empleara Node-Red para comunicarse y enviar las instrucciones a la base de datos. Para ello hay que realizar un nuevo usuario como se ve en la imagen 47, donde se debe de añadir la IP donde trabaja el usuario, en este proyecto se colocara la IP de la Revolution Pi y una contraseña de seguridad. Este usuario y contraseña serán los que se colocan en las propiedades del nodo MySQL de Node-RED comentado anteriormente en la imagen 36.



User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
usercaamiones	192.168.1.75

Details for account usercaamiones@192.168.1.75

Login Name: usercaamiones

Authentication Type: Standard

Limit to Hosts Matching: 192.168.1.75

Password: [masked]

Confirm Password: [masked]

Expire Password

Imagen 47. Creación de un nuevo usuario para trabajar con la base de datos desde otra IP.

A parte de eso también debe de indicar los privilegios que poseerá este usuario a la hora de trabajar con dicha base de datos, ya que si no permites que el usuario pueda escribir o sacar datos tampoco funcionara el programa. Estos privilegios que debe de permitir el usuario son los que se observa en la imagen 48 y en este proyecto son todos, ya que así tienes el mando de la base en la Revolution Pi.

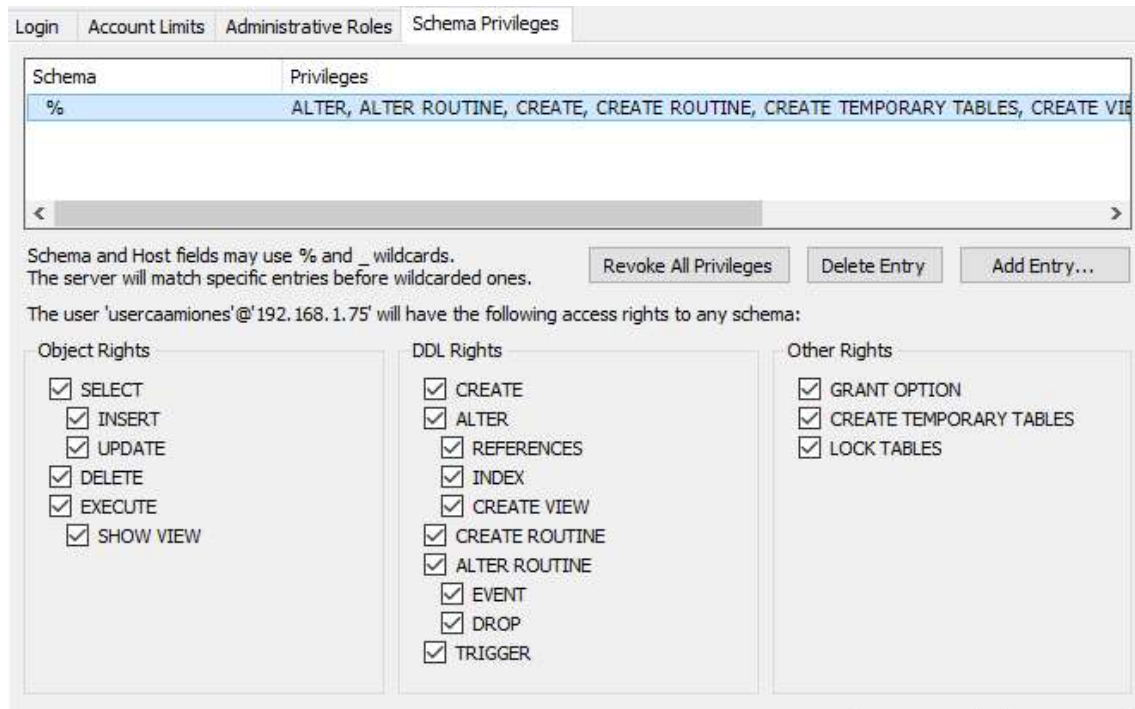


Imagen 48. Privilegios otorgados al usuario para trabajar con la base de datos.

10. Planificación.

En este apartado del proyecto se va a detallar paso por paso como se ha planificado la realización del proyecto desde que se presentó el proyecto hasta la finalización de este.

Lo primero que se realizó por parte de la empresa fue presentar el proyecto al ingeniero comentando todas las partes de este. También se le indico al ingeniero un ejemplo de cómo debería de ser el funcionamiento del proyecto a crear.

Una vez el ingeniero asimilo todos los datos indicados por la empresa empezó a organizar las ideas del proyecto e ir conociendo los elementos importantes del proyecto como son el ordenador de placa reducida empleado en el proyecto o los programas de realización de programas de automatización como CODESYS o el programa para crear y trabajar bases de datos como MySQL Workbench.

Cuando ingeniero se familiarizo con los programas anteriores, este presento una idea de cómo realizar el proyecto a la empresa, a los cuales le parecieron correctos estas ideas. A su vez desde la empresa se le entrego al proyectista los manuales de los elementos que ha seleccionado la empresa para este proyecto como las células de carga, la cámara o la barrera automática que se deberán de emplear.

Después de poner todos los datos en común el proyectista empezó a ver como poder comunicar los datos que se obtienen de los dispositivos anteriores con el proyecto que se encuentra en la Revolution Pi como la cámara que se comunicará con el mini PC vía Ethernet o la barrera que será mediante una señal digital.

Una vez visto todas las conexiones se observó que el transmisor de peso analógico que se emplea para comunicar se comunica con la caja de conexiones, la cual sirve para enviar los pesos obtenidos de las celdas al transmisor, recibe esa señal mediante conexión RS485 y posteriormente envía el peso de forma analógica. Pero se observó que la Revolution Pi puede recibir señales vía RS485 y la cual es una transmisión mejor que la analógica y por lo tanto se quitó del proyecto el transmisor de peso analógico.

Por último, ya se realizó el programa y en el cual se determinó el empleo del programa Node-RED para comunicar el programa realizado con CODESYS con la base de datos, ya que la librería que emplea CODESYS para comunicarse con MySQL es pagando, mientras que la comunicación mediante Node-RED es gratuita.

Cuando ya se realizó el programa este se presentó a la empresa explicando cómo se realiza y los elementos que se han empleado para ello con sus respectivas comunicaciones con el programa. Una vez realizado dicha presentación la empresa dio el visto bueno y entonces se empezó a redactar el proyecto.

11. Resumen de presupuesto.

En este apartado se va a resumir al documento de este proyecto donde se habla del gasto que hay que asumir en este proyecto. Cabe indicar que al presupuesto de este proyecto habría que sumarle el coste de la construcción de la báscula como ya se ha comentado en el apartado de diseño de la báscula.

El presupuesto de este proyecto se ha separado en dos apartados diferentes. En el primer apartado se calculará el precio del material que se necesita para la realización del proyecto. En el segundo apartado del presupuesto se obtendrá el coste por el tiempo que ha empleado el ingeniero en realizar todo el proyecto, desde el conocimiento de todos los programas empleados, la comunicación entre los dispositivos y la creación del programa que poseerá el autómatas.

Por último se han sumado ambos presupuestos para obtener el presupuesto de ejecución del material al cual posteriormente se le suma un porcentaje de este empleado para los gastos generales y el beneficio industrial y con el cual obtenemos el presupuesto de ejecución por contrata y al cual solo se le deberá sumar los honorarios del proyectista y 21% correspondiente a los impuestos para obtener el presupuesto final, el cual asciende a un total de quince mil novecientos veintiocho con veinticuatro euros.

12. Conclusión.

En conclusión, en este proyecto se ha diseñado e implementado el programa de automatización para un ordenador de placa reducida que permite controlar el acceso de entrada de camiones cumpliendo las especificaciones planteadas.

Hay que destacar que se trata de un proyecto de automatización complejo debido al empleo de un nuevo lenguaje de programación diferente a los lenguajes estudiados en el grado e indicados en la norma IEC 61131 como ha sido Node-RED. Por lo que se ha debido de investigar desde un principio como trabajar con este programa y cómo realizar la vinculación de este con el programa principal realizado con CODESYS mediante los Modbus.

El programa general realizado para este proyecto se ha realizado empleando diferentes lenguajes de programación según la facilidad que da cada uno de estos lenguajes. Mientras que con Node-RED se ha creado el programa que permite la comunicación de los datos obtenidos con los de la base de datos.

Por otro lado, también se ha tenido que realizar una búsqueda de los elementos hardware seleccionados para llevar a cabo este proyecto y que mejores poseen las mejores condiciones y más facilidades dan para realizar las comunicaciones entre ellos y el ordenador de placa reducida. También cabe indicar que al tratarse de un proyecto para la instalación en zonas exteriores los dispositivos seleccionados deben de tener una protección contra los efectos exteriores, lo que provoca un aumento del precio de los elementos y por tanto del proyecto.

13. Bibliografía.

Se han empleado apuntes de las siguientes asignaturas realizadas en la carrera:

- ET1031 – Proyectos de Ingeniería.
- ET1030 – Automatización Industrial.
- ET1032 – Informática Industrial.
- ET1023 – Sistemas Automáticos.
- ET1018 – Electrónica.

Además de estas asignaturas, también se han empleados los siguientes trabajos de final de grado que se encuentran en la biblioteca de la universidad:

- *“Modelización de un brazo móvil a partir de un cabrestante para su uso en vehículos o remolques como ayuda de carga.”* de Lledó Fortea Pallarés.
- *“Diseño de una máquina para la separación del fruto recolectado del envase utilizado en dicha actividad, en el proceso de industria hortofrutícola.”* de Javier Pérez Mengual.
- *“Actualización y mejora de la automatización de un horno para esmaltes”* de Javier Albalade Escorihuela.

A parte de la información que se ha obtenido de todos los documentos anteriores, también se ha buscado información relacionada con los programas empleados y los elementos hardware del proyecto en las siguientes páginas webs y en las fichas de datos de dichos dispositivos empleados que se encuentran en el apartado de los anexos.

- CODESYS

<https://larraioz.com/codesys>

https://larraioz.com/codesys?utm_source=google_ads&utm_medium=cpa&utm_campaign=codesys022019&&keyword=codesys&device=c&network=g&gclid=CjwKCAjwlbr8BRA0EiwAnt4MTs4x0SpRCMzDmRhHimWbYg4v-7zQlativ7fgViXHuzQgawiSUnE3BoCILIQA vD_BwE

- Node-RED

<https://nodered.org/>

<https://www.toptal.com/nodejs/programacion-visual-con-node-red-conectando-el-internet-de-las-cosas-con-facilidad>

<https://iotconsulting.tech/que-es-node-red-y-para-que-sirve/>

- My SQL

<https://www.mysql.com/>

<https://es.wikipedia.org/wiki/MySQL>

- Revolution Pi RevPi Core 3

<https://revolution.kunbus.com/revpi-core/>

<https://www.pilz.com/es-ES/eshop/00103002137157/Revolution-Pi>

- Revolution Pi DIO

<https://revolution.kunbus.com/io-modules/>

<https://revolution.kunbus.de/shop/en/hardware>

- Revolution Pi - USB-RS485

<https://revolution.kunbus.de/shop/en/usb-rs485-konverter>

- Celdas de carga bascula

<https://www.laumas.com/es/producto/cbl-celulas-de-carga-de-compresion-perfil-bajo/>

<https://spanish.alibaba.com/product-detail/cbl-compression-load-cell-stainless-steel-17-4-ph-capacity-from-250-kg-to-100000-kg-50033200648.html>

- Caja de conexiones

<https://www.laumas.com/es/producto/clm8-cajas-de-conexiones-inteligentes-8-16-canales-para-celulas-de-carga/>

<https://www.plc-city.com/shop/es/conexiones-y-cableados/clm8-nfs.html>

<https://www.plc-city.com/shop/es/electronica-de-pesaje/castlguafnfs.html>

- Cámara

<https://www.hikvision.com/en/products/IP-Products/Network-Cameras/DeepinView-Series/DS-2CD7A26G0-P-IZ-H-S/>

http://www.ciudadwireless.com/hikvision_solutions_ds-2cd7a26g0-p-izhs-2-8-12mm-deepinview_camera-2-8-12mm_motorized_vari-focal_darkfighter-p-19586.html

- Barrera

<https://www.aprimatic.es/producto/barreras-automaticas-3/areas-residenciales-barreras-automaticas-3/barrera-automatica-park-30-plus/>

<https://www.webdosb.com/barreras/6242-barrera-aprimatic-park-24v-de-3m.html>

- Fotocélula

<https://es.rs-online.com/web/p/sensores-fotoelectricos/6956833/>

<https://es.rs-online.com/web/p/accesorios-para-sensores-fotoelectricos/2811751/?sra=pmpn>

- Sensor inductivo

<https://es.rs-online.com/web/p/sensores-de-proximidad-inductivos/2010785/?sra=pmpn>

Anexos



Índice anexos.

1. Programa CODESYS.....	63
1.1. Lista de variables globales: GVL_proyecto	63
1.2. POU: FLANCO_PULSA	64
1.3. PLC_PRG.....	65
1.4. POU: PROGRAMA_COM	67
1.5. POU: PROGRAMA_FALLOS_COM.....	69
1.6. PROGRAMA_FALLOS_REG.....	69
1.7. POU: PROGRAMA_REG.	71
1.8. POU: PROGRAM_TEMP.....	72
2. Programa Node-RED.....	73
2.1. Programa de obtención y guardado de las variables en Node-RED.	73
2.2. Programa para registrar los datos.	74
2.3. Programa de comparar y reiniciar los datos.....	75
3. Catálogos de los elementos del proyecto.....	77
3.1. Ficha de datos de la Revolution Pi RevPi Core 3.	77
3.2. Ficha de datos de la RevPi DIO.....	78
3.3. Ficha de datos de las células de carga.	80
3.4. Ficha de datos de la caja de conexiones.	81
3.5. Ficha de datos de la cámara HIKVISION.....	82
3.6. Ficha de datos de la placa TRAFFIC VF de la barrera automática.	87
3.7. Ficha de datos del detector de lazo inductivo.	93
3.8. Ficha de datos de la fotocélula réflex polarizada.....	100
3.9. Ficha de datos del semáforo y la placa de conexión para el semáforo.	104

1. Programa CODESYS.

1.1. Lista de variables globales: GVL_proyecto

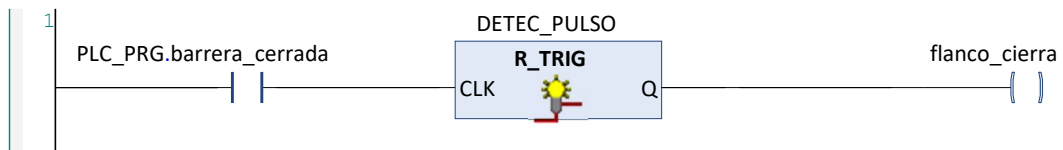
```

1      {attribute 'qualified_only'}
2      VAR_GLOBAL
3          //Variables de registro de los datos.
4          matricula : STRING ( 16 ) ;
5          dni : STRING ( 16 ) ;
6          peso : UINT ;
7          id : UINT ; 8
9          //Variables entrada Node-red.
10         reinicia AT %QW14 : BOOL ;
11         matricula_node AT %QW0 : STRING ( 16 ) ;
12         dni_node AT %QW4 : STRING ( 16 ) ;
13         peso_node AT %QW9 : UINT ;
14         id_node AT %QW13 : UINT ;
15         envia_compara AT %QW12 : BOOL ;
16         envia_registra AT %QW11 : BOOL ;
17
18         //Variables salida Node-red
19         id_bbdd AT %IW3 : UINT ;
20         peso_bbdd AT %IW4 : UINT ;
21         peso_dni_bbdd AT %IW5 : UINT ;
22         peso_matri_bbdd AT %IW6 : UINT ;
23         compara_correctos AT %IW2 : BOOL ;
24
25         comparacion_correcta : BOOL ; //Variable BOOL que activa la
26         transición y que indica que los datos son correctos.
27         registrado : BOOL ; //Variable BOOL que activa la transición y
28         que indica que los datos son correctos.
29         datos_correcto : BOOL ; //Variable que activa la luz para
30         cuando son correctos.
31         com_peso : BOOL ; //Variable que indica que el peso está dentro
32         del rango de error.
33
34         registra : BOOL ; //Pulsador para iniciar el registro.
35         compara : BOOL ; //Pulsador para iniciar la comparación.
36         pulsa_restaurar : BOOL ; //Pulsador por si se quiere cambiar de
37         registrar a comparar.
38
39         //Errores
40         error_dni : BOOL ; //Fallo en la toma de DNI.
41         error_matri : BOOL ; //Fallo en la toma de matrícula.
42         error_peso : BOOL ; //Fallo en la toma de peso.
43
44         restaurado : BOOL ; //Variable de transición para que los
45         datos se han restaurado.
46     END_VAR

```

1.2. POU: FLANCO_PULSA

```
1  PROGRAM FLANCO_PULSA
2  VAR
3      flanco_cierra : BOOL ;
4      DETEC_PULSO : R_TRIG ;
5  END_VAR
6  //Programa corto que lo único que hace es determinar si el pulsador reinicia se ha activado, es decir,
7  //detecto un flanco de subida lo que activara la señal interna para reiniciar el programa.
8
```

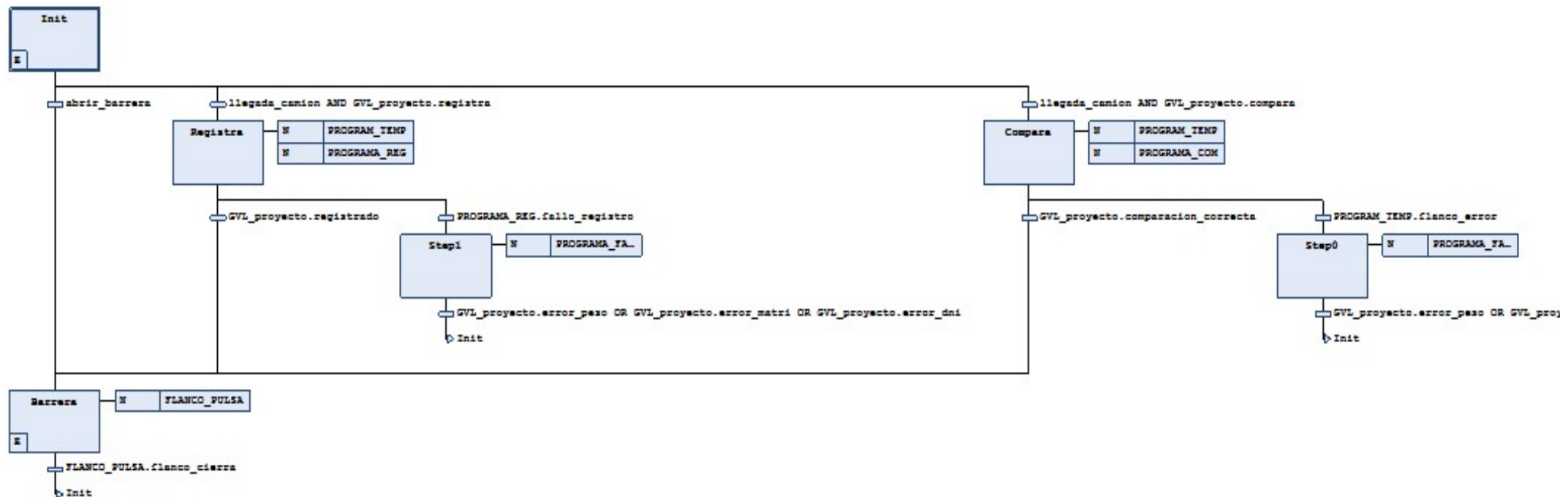


1.3. PLC_PRG.

```

1  PROGRAM PLC_PRG
2  VAR
3      llegada_camion : BOOL ; //Sensor de detección de la llegada del
    transporte.
4      barrera_cerrada : BOOL ; //Sensor que indica que la barrera esta cerrada
5      open_barrera : BOOL ; //Señal de salida para que se activa la barrer.
6      abrir_barrera : BOOL ; //Pulsador parra abrir la barrera en caso de
    emergencia.
7  END_VAR
8

```



1.3.1. Acción: Barrera_entry.

```

1      //Acción de entrada que lo único que hace es activar la señal
      para abrir 1 barrera y desactiva las señales de que los datos se
      han recibido.
2      open_barrera := TRUE ;
3      GVL_proyecto . datos_correcto := FALSE ;
4      GVL_proyecto . registrado := FALSE ;
5      GVL_proyecto . comparacion_correcta := FALSE ;
7

```

1.3.2. Acción: start.

```

1      //Programa de activación a la entrada.
2      //Reinicia todas las variables a valores iniciales.
3      GVL_proyecto . matricula := ' ' ;
4      GVL_proyecto . dni := ' ' ;
5      GVL_proyecto . peso := 0 ;
6      GVL_proyecto . id := 0 ;
7      GVL_proyecto . matricula_node := GVL_proyecto . matricula ;
8      GVL_proyecto . dni_node := GVL_proyecto . dni ;
9      GVL_proyecto . id_node := GVL_proyecto . id ;
10     GVL_proyecto . peso_node := GVL_proyecto . peso ;
11
12
13
14     GVL_proyecto . com_peso := FALSE ;
15
16     GVL_proyecto . datos_correcto := FALSE ;
17     PLC_PRG . open_barrera := FALSE ;
18     GVL_PROYECTO . compara := FALSE ;
19     GVL_proyecto . comparacion_correcta := FALSE ;
20     GVL_proyecto . envia_compara := FALSE ;
21     GVL_proyecto . registrado := FALSE ;
22     GVL_proyecto . registra := FALSE ;
23     GVL_proyecto . restaurado := FALSE ;
24     GVL_proyecto . reinicia := FALSE ;

```

1.4. POU: PROGRAMA_COM

```

1      PROGRAM PROGRAMA_COM
2      VAR
3
4      END_VAR
5

```

```

1      //Copia los datos introducidos de las variables en las
      //variables que se encuentran comunicadas con el
2      //programa de Node-RED.
3      GVL_proyecto . matricula_node := GVL_proyecto . matricula ;
4      GVL_proyecto . dni_node := GVL_proyecto . dni ;
5      GVL_proyecto . id_node := GVL_proyecto . id ;
6      GVL_proyecto . peso_node := GVL_proyecto . peso ;
7
8      //Cuando se pulsa para que inicie la comparación enviara una señal
      //para que empiece el temporizador.
9      GVL_proyecto . envia_compara := PROGRAM_TEMP . haz_com ;
10
11
12      //Reinicia los datos de fallos a FALSE
13      GVL_proyecto . error_peso := FALSE ;
14      GVL_proyecto . error_matri := FALSE ;
15      GVL_proyecto . error_dni := FALSE ;
16
17      //Como el peso de llegada y el que se encuentra registrado en
      //Node-RED es muy probable que no sea el mismo se ha indicado un
      //redondeo de valores para que autorice el peso. Por lo que de la
      //bbdd se obtiene el peso guardado y se compara con el introducido
      //ahora y si el error es +-5% el dato es
18      //correcto.
19      IF GVL_proyecto . peso_node >= 0.95 + GVL_proyecto . peso_bbdd
      AND GVL_proyecto
      . peso_node <= 1.05 + GVL_proyecto . peso_bbdd THEN
20          GVL_proyecto . com_peso := FALSE ;
21      ELSE
22          GVL_proyecto . com_peso := TRUE ;
23      END_IF
24
25
26      //Como no he podido sacar los datos para que se comparen, he
      //obtenido el peso con cada uno de los
27      //datos introducidos y se han comparado entre sí.
28
29
30      //Si estos datos son correctos se reinician los valores a 0 que se
      //envían a Node-RED y se activan
31      //las señales de que los datos son correctos y se desactiva la
      //señal para que compare.
32      IF GVL_proyecto . peso_dni_bbdd = GVL_proyecto .
      peso_matri_bbdd AND GVL_proyecto . peso_dni_bbdd = GVL_proyecto .
      peso_bbdd AND
      GVL_proyecto . compara_correctos = TRUE AND GVL_proyecto .
      com_peso = FALSE THEN
33          GVL_proyecto . id_node := 0 ;
34
35          GVL_proyecto . peso_node := 0 ;
36          GVL_proyecto . matricula_node := ' ' ;

```

```
36         GVL_proyecto . dni_node := ' ' ;
37         GVL_proyecto . comparacion_correcta := TRUE ;
38         GVL_proyecto . datos_correcto := TRUE ;
39         GVL_proyecto . compara := FALSE ;
40         GVL_proyecto . reinicia := TRUE ;
41     END_IF
```

1.5. POU: PROGRAMA_FALLOS_COM

```

1  PROGRAM PROGRAMA_FALLOS_COM
2  VAR
3  END_VAR
4


---


1  //Sirve para el mismo proceso que el programa PROGRAMA_FALLOS_REG
   lo único que este compara el
2  //peso obtenido solo por el id con el obtenido con el id y la
   matrícula o el dni. Ya que
3  //si el dni o la matrícula no corresponde con el id el dato que
   obtenemos del Node-RED es 0.
4  IF GVL_proyecto . peso_bbdd = GVL_proyecto . peso_matri_bbdd THEN
5      GVL_proyecto . error_matri := FALSE ;
6  ELSE
7      GVL_proyecto . error_matri := TRUE ;
8  END_IF
9
10 IF GVL_proyecto . peso_bbdd = GVL_proyecto . peso_dni_bbdd
   THEN
11     GVL_proyecto . error_dni := FALSE ;
12 ELSE
13     GVL_proyecto . error_dni := TRUE ;
14 END_IF
15
16 //En este caso si al realizar la comparación esta ha sido errónea
   activara la señal de fallo.
17 IF GVL_proyecto . com_peso = FALSE THEN
18     GVL_proyecto . error_peso := FALSE ;
19 ELSE
20     GVL_proyecto . error_peso := TRUE ;
21 END_IF
22

```


1.6. PROGRAMA_FALLOS_REG

```
1  PROGRAM PROGRAMA_FALLOS_REG
2  VAR
3  END_VAR
4


---


1  //Programa simple que lo único que hace es ver que variable a
2  fallado y activa la señal visual.
3  IF GVL_proyecto . matricula = ' ' THEN
4      GVL_proyecto . error_matri := TRUE ;
5  ELSE
6      GVL_proyecto . error_matri := FALSE ;
7  END_IF
8
9  IF GVL_proyecto . dni = ' ' THEN
10     GVL_proyecto . error_dni := TRUE ;
11 ELSE
12     GVL_proyecto . error_dni := FALSE ;
13 END_IF
14
15 IF GVL_proyecto . peso = 0 THEN
16     GVL_proyecto . error_peso := TRUE ;
17 ELSE
18     GVL_proyecto . error_peso := FALSE ;
19 END_IF
```

1.7. POU: PROGRAMA_REG.

```

1  PROGRAM PROGRAMA_REG
2  VAR
3      registra_correcto AT %IW1: BOOL ;
4
5      fallo_registro : BOOL ;
6  END_VAR

1  //Copia los valores de entrada a las variables que estan en
   //comunicación con el programa Node-RED
2  GVL_proyecto . matricula_node := GVL_proyecto . matricula ;
3  GVL_proyecto . dni_node := GVL_proyecto . dni ;
4  GVL_proyecto . peso_node := GVL_proyecto . peso ;
5
6  //Restaura a valores iniciales las variables que indican un
   //error
7  GVL_proyecto . error_peso := FALSE ;
8  GVL_proyecto . error_matri := FALSE ;
9  GVL_proyecto . error_dni := FALSE ;
10 fallo_registro := FALSE ;
11
12 //Realiza si los valores que se han introducido son correctos a
   //alguno de ellos no se a introducido.

13 //En este segundo caso , se activara la señal de de hay un
   //error en el momento de introducir los
14 //datos. Si los datos son correctos se activara la señal que
   //active el registro.
15 IF GVL_proyecto . matricula = ' ' OR GVL_proyecto . dni = ' ' OR
   GVL_proyecto . peso = 0 THEN
16     fallo_registro := TRUE ;
17 ELSE
18     GVL_proyecto . envia_registra := PROGRAM_TEMP . has_registro ;
19
20 END IF
21
22 //Cuando desde Node-RED se recibe una señal que indica que se ha
   //realizado el registro se obtendra
23 //el valor del ID donde se ha guardado el dato se activaran las
   //variables de activación y se
24 //desactivaran las variables de envio de datos para que no se siga
   //enviando datos a Node-RED.
25 IF registra_correcto = TRUE THEN
26
27     GVL_proyecto . id := GVL_proyecto . id_bbdd ;
28     GVL_proyecto . registrado := TRUE ;
29     GVL_proyecto . datos_correcto := TRUE ;
30     GVL_proyecto . registra := FALSE ;
31     GVL_proyecto . envia_registra := FALSE ;
32 END_IF
33

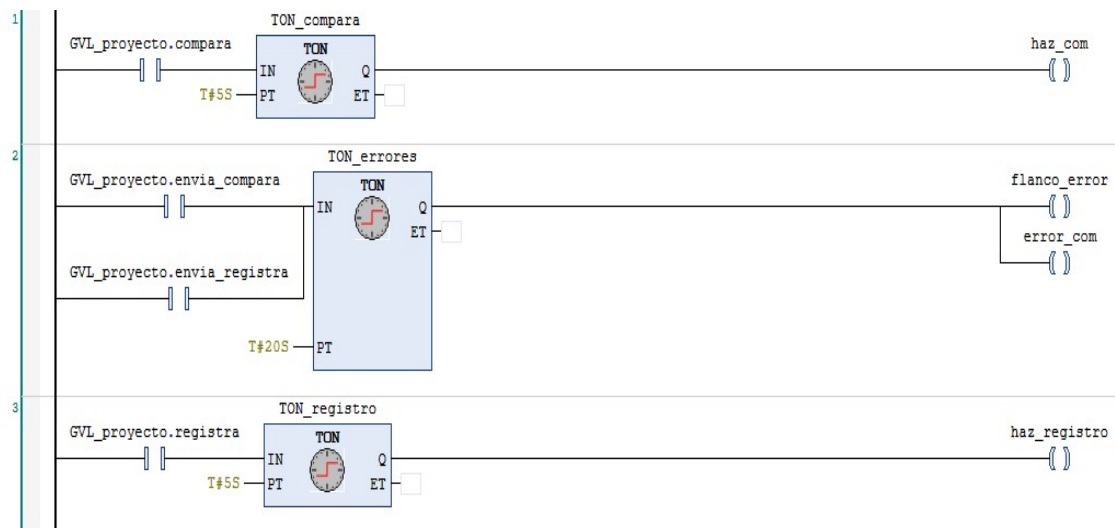
```

1.8. POU: PROGRAM_TEMP.

```

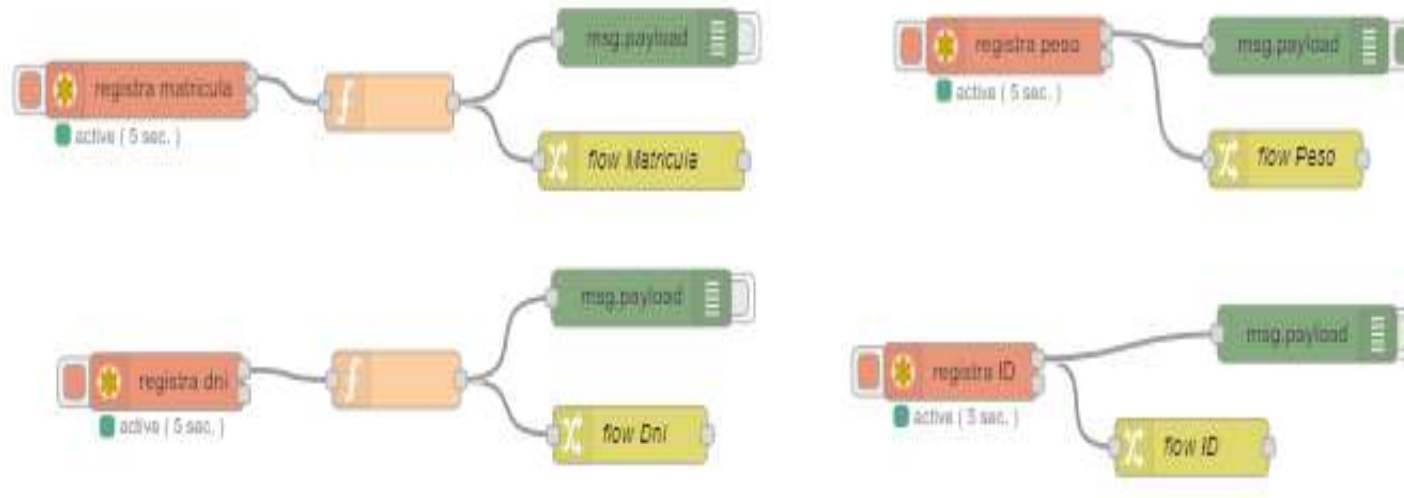
1  PROGRAM PROGRAM_TEMP
2  VAR
3
4      error_com : BOOL ;
5      TON_compara : TON ;
6      haz_com : BOOL ;
7      flanco_error : BOOL ;
8      haz_registro : BOOL ;
9      TON_registro : TON ;
10     TON_errores : TON ;
11 END_VAR
12
13 //Red 1 y Red 3 hacen el proceso que espera 5s a que el Node-
14 RED halla registrado ya
15 //los datos de las variables para que no surjan errores al
16 //comparar o registrar.
17
18 //Red2: Si no ha habido datos recibidos desde Node-RED en un
19 //determinado tiempo activa
20 //señal de que hay algún fallo.

```

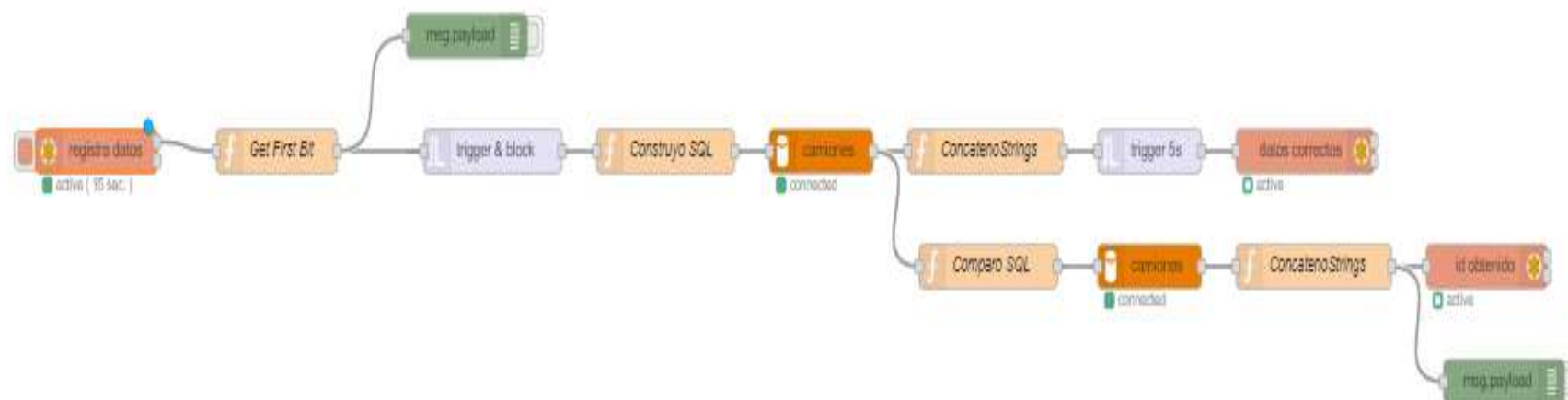


2. Programa Node-RED.

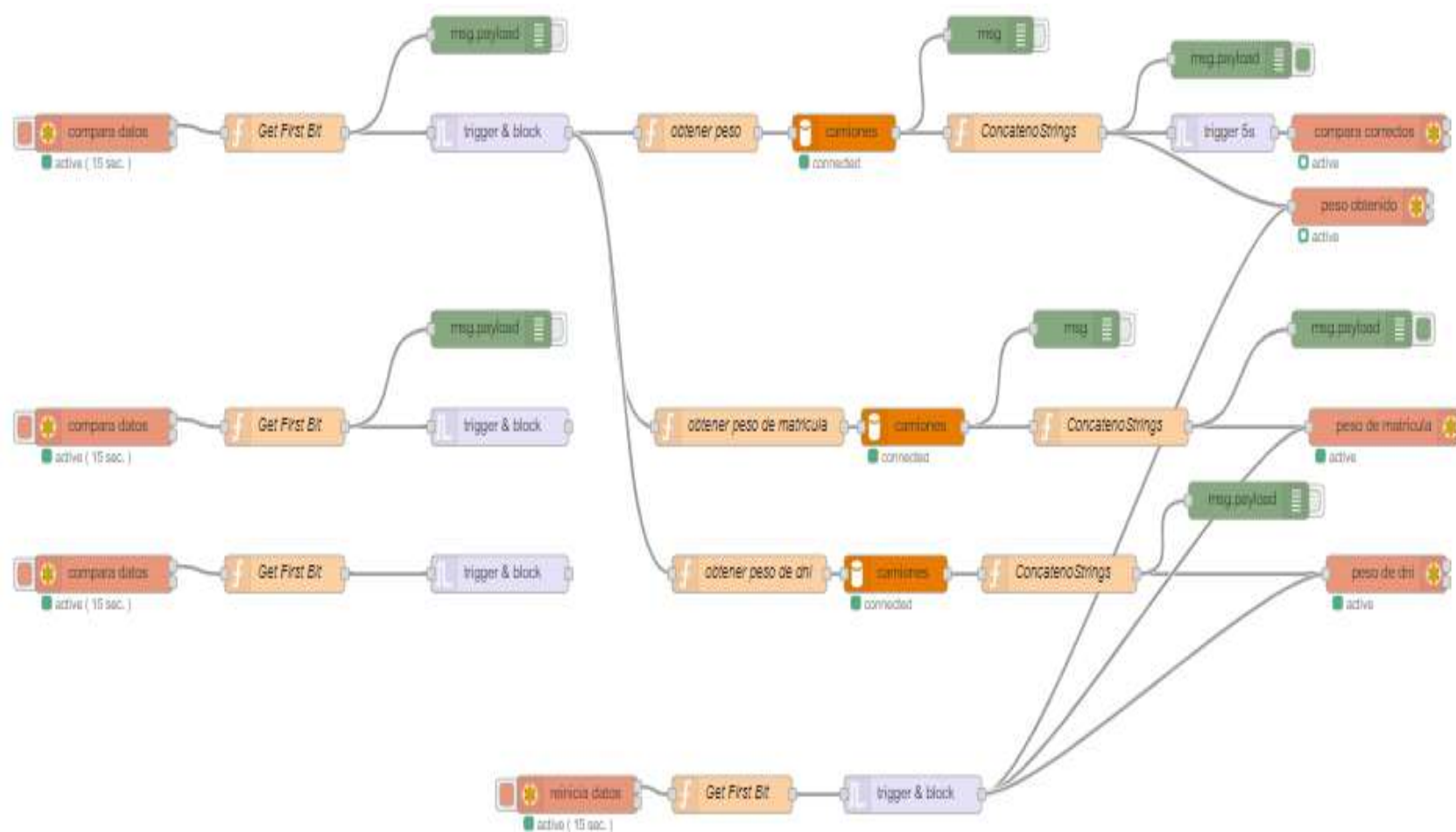
2.1. Programa de obtención y guardado de las variables en Node-RED.



2.2. Programa para registrar los datos.



2.3. Programa de comparar y reiniciar los datos.



3. Catálogos de los elementos del proyecto.

3.1. Ficha de datos de la Revolution Pi RevPi Core 3.

REVOLUTION PI

RevPi Core 3

Article no.: 100257



Technical Data

Housing dimensions (HxWxD)	96 x 22.5 x 110.5 mm
Housing type	DIN rail housing (for DIN rail version EN 50022)
Housing material	Polycarbonate
Weight	approx. 115 g
Protection class	IP20
Power supply	12-24 VDC -15 % / +20 %, reverse polarity protected ¹
Max. power consumption	10 W (incl. 900 mA total USB load) ²
Operating temperature	-40 °C to +55 °C ³
Storage temperature	-40 °C to +85 °C
Humidity (40 °C)	Up to 93 % (non-condensing)
Interfaces	2 x USB A (Total current consumption from both sockets max. 900 mA) ² 1 x RJ45 10/100 Ethernet 1 x Micro-USB (solely for image transfer to eMMC) 1 x Micro HDMI 2 x PiBridge system bus
Connectors	1 x 4-pole screw-type terminal for power supply
Processor	Broadcom BCM2837 quad-core ARM Cortex-A53
Clock rate	1.2 GHz
Processor cooling	Passive with heat sink
RAM	1 GB
Flash memory	4 GB
Compatible modules for system expansion	All RevPi IO modules and RevPi Gate modules can be connected via the PiBridge system bus
ESD protection	4 kV / 8 kV (according to EN 61131-2 and IEC 61000-6-2)
EMI tests	Passed (according to EN 61131-2 and IEC 61000-6-2)
Surge / Burst tests	Passed (according to EN 61131-2 and IEC 61000-6-2)
Buffer time RTC	min. 24 h
Optical indicator	3 status LEDs (bi-color), two of them freely programmable
Conformity	CE, RoHS
UL certification	Yes, UL-File-No. E494534

¹ 900 mA USB output current (sum of both USB outputs) is only available at input voltages >11 V. The bridging time required by EN 61131-2 of voltage dips of at least 10 ms is only guaranteed with a supply voltage of 20.4 to 28.8 V. At 12 V input voltage this time decreases drastically, especially when driving loads by USB ports.

² The average power consumption without USB loads vary widely and depends on the specific use of interfaces, GPU and CPU. Not using the HDMI interface keeps the power consumption of generally below 4 W.

³ There should be no cutbacks of compute power at ambient temperatures under 20°C. At 25°C ambient temperature 3 cores may run with full clock speed while with 4 cores the clock frequency is lowered from 1.2 to 1.1 GHz after 10 to 20 minutes of full stress. At 40°C ambient temperature 4 cores under full stress will still work with 1 GHz while stressing just 1 core results in no down clocking. At 50°C ambient temperature 4 fully stressed cores are running at average 0.7 GHz, having short down clockings to 0.6 GHz and short up clockings to 0.9 GHz, 1 core under full stress does result in no down clocking. At 65°C ambient temperature and either 4 or 1 core under full stress results in an "emergency mode" with just 0.4 GHz, after longer periods even 0.3 GHz.

3.2. Ficha de datos de la RevPi DIO.

REVOLUTION PI

RevPi DIO

Article No.: 100197



Technical Data

Housing dimensions (H x W x D)	96 x 22.5 x 110.5 mm
Housing type	DIN rail housing (for DIN rail version EN 50022)
Housing material	Polycarbonate
Weight	approx. 100 g / 130 g (incl. connectors)
IP Code	IP20
Power supply	12-24 VDC -5 % / +20 % (X2 and X4) ¹
Max. power consumption	1.5 Watt (X4/power supply)
Operating temperature	-40 °C...+55 °C
Storage temperature	-40 °C...+85 °C
Humidity (at 40 °C)	93 % (non-condensing)
Connectors	2 x 4-pole screw-type terminal for power supply 2 x 14-pin socket connectors with spring clamp contacts (0.2 - 1.5 mm ²) for IOs, pitch 3.5 mm (Wieland Item No. 27.630.4453.0)
Optical indicator	3 status LEDs (bi-color)
Number of digital input channels	14
Input type	Galvanically isolated from the system bus and from the outputs, individually configurable as direct digital input, counter rising edge, counter falling edge or together with neighbored input as encoder ²
Input current limitation	2.4 mA (at 24 V power supply)
Input thresholds	At 24 V compatible according to EN 61131-2 to Type I and III sensors
Digital debounce circuit	Collectively adjustable for all inputs: off, 25 µs, 750 µs or 3 ms
Maximum frequency resolution of the counter inputs	2 kHz (corresponding to 500 Hz encoder sequence)
Alarm	For auxiliary voltages below 19 V and below 9 V, overtemperature
Input protection	According to EN 61131-2 (IEC 61000-4-4, -5, -6, and -2) against burst, RF injection, external voltages from -45 V to +45 V

¹ Three independent supply voltage sources must be available for galvanic isolation of the inputs and outputs.

² For each DIO module, a maximum of 6 inputs can be defined as 6 counters or 12 inputs as 6 decoders. Counters and decoders are stored as 32-bit integers in the process image. Reset of counters/encoders via ioct() calls from the kernel driver piControl.

REVOLUTION PI

RevPi DIO

Article No.: 100197

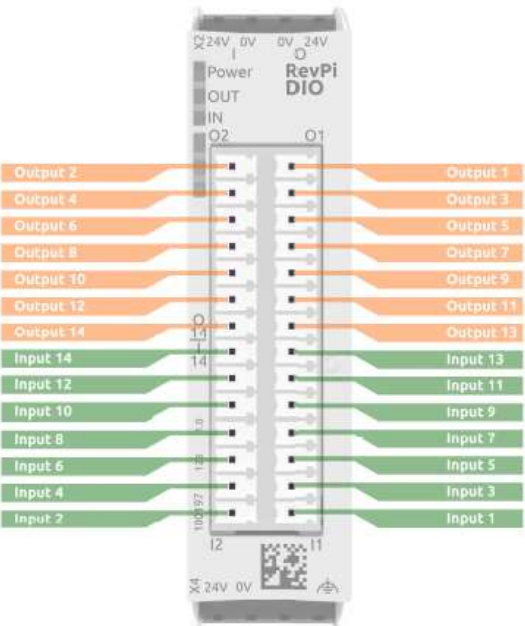
Technical Data

Number of digital output channels	14
Output type	Galvanically isolated from the system bus and the inputs, individually configurable as direct digital output with high-side or push-pull drivers as well as a PWM output ³
Maximum current per output	500 mA (high-side mode), 100 mA (push-pull mode)
PWM frequency	Collectively selectable for all outputs: 40 Hz, 80 Hz, 160 Hz, 200 Hz, 400 Hz ³
Alarm	Thermal shutdown or short circuit of outputs (individually for each output)
Dual watchdog function	In the event of communication failure with the controller (after 50 ms) or internal communication with the CPU (after 9 ms, hardware-controlled), the outputs are reseted to zero
Output protection	According to EN 61131-2 (IEC 61000-4-4, -5, -6, and -2) against short circuit, overload, burst, ESD
Compatible modules for system expansion	All RevPi base module, expansion modules and RevPi Gate modules (connected via overhead PiBridge connector)
Protection of the power supply inputs	Reverse polarity protected, overvoltage protection
Markings and certifications	CE, UL
RoHS conformity	Yes
Surge/Burst tests	Passed (according to EN 61131-2 and IEC 61000-6-2)

³ The PWM pulse width is stored as a value from 0 to 100 in the process image in 1 byte. The maximum resolution of the conversion of this value in % by the module depends on the PWM Frequency: 40 Hz / 1%, 80 Hz / 2%, 160 Hz / 4%, 200 Hz / 5%, 400 Hz / 10%.

RevPi DIO

Item no.: 100197



3.3. Ficha de datos de las células de carga.

CARACTERÍSTICAS TÉCNICAS

Material		Acero inoxidable 17-4 PH		
Clase de precisión OIML R60 • Divisiones de comprobación de la escala		-	C2 • 2000	C3 • 3000
Carga nominal (E max)		250 - 500 - 1000 - 15000 kg 30000 - 50000 - 100000 kg	2500 - 5000 kg 7500 - 10000 kg	2500 - 5000 - 7500 kg 10000 - 12500 kg
Escalón mínimo de verificación (V min)		■	E max / 15000	E max / 15000
Error combinado		$\leq \pm 0,03\%$	$\leq \pm 0,03\%$	$\leq \pm 0,02\%$
Grado de protección		IP68		
Sensibilidad	2 mV/V $\pm 0,1\%$	Resistencia de entrada		700 $\Omega \pm 10$
Efecto de la temperatura en cero	0,005% °C	Resistencia de salida		700 $\Omega \pm 10$
Efecto de la temperatura en el fondo de escala	0,003% °C	Balance en cero		$\pm 1\%$
Compensación térmica	-10 °C / +50 °C	Resistencia de aislamiento		>10000 M Ω
Rango de temperatura de trabajo	-20 °C / +70 °C	Carga estática máxima (% en el fondo de escala)		150%
Fluencia en carga nominal después de 30 minutos	0,03%	Carga de rotura (% en el fondo de escala)		300%
Tensión de alimentación máxima tolerada	15 V	Deflexión con carga nominal		0,4 mm

CONEXIONES ELÉCTRICAS

Longitud de cable	5 m* (250-10000 kg); 10 m (12500-100000 kg)
Diámetro del cable	5 mm
Hilos conductores	4 x 0,25 mm ² /6 x 0,14 mm ²

*j) Bajo pedido: versión cable 10 m



3.4. Ficha de datos de la caja de conexiones.

CARACTERÍSTICAS TÉCNICAS

Alimentación y potencia absorbida	12÷24 VDC ±10%; 5 W
Número de células de carga • Alimentación de las células de carga	hasta 16 (350 Ω) - 4/6 hilos • 5 VDC/240 mA
Linealidad	<0.01% fondo de escala
Deriva térmica	<0.0005% fondo de escala/°C
Convertidor A/D	8 canales - 24 bit (16000000 puntos) - 4.8 kHz
Divisiones (rango de medición ±10 mV y sensibilidad 2 mV/V)	±999999 • 0.01 µV/d
Rango de medición	±39 mV
Sensibilidad células de carga empleables	±7 mV/V
Conversiones por segundo	600/s
Rango visualizable	±999999
Número decimales • Resolución de lectura	0÷4 • x1 x2 x5 x10 x20 x50 x100
Filtro digital • Lecturas por segundo	21 niveles • 5÷600 Hz
Puertos serie	RS485, RS232
Velocidad de transmisión	2400, 4800, 9600, 19200, 38400, 115200 (bit/s)
Humedad (no condensante)	85%
Temperatura de almacenaje	-30 °C +80 °C
Temperatura de trabajo	-20 °C +60 °C



Utilizar una fuente de alimentación externa 12-24 VDC de tipo LPS o en clase 2.

CARACTERÍSTICAS METROLÓGICAS DE LOS INSTRUMENTOS HOMOLOGADOS

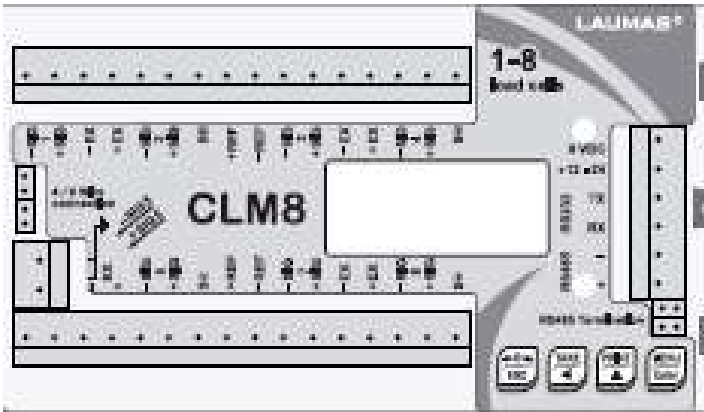
Normas respetadas	2014/31/UE - EN45501:2015 - OIML R76:2006
Modos de funcionamiento	rango único, intervalo múltiple, rango múltiple
Clase de precisión	III o IIII
Número máximo de divisiones de comprobación de la escala	10000 (clase IIII); 1000 (clase IIII)
Número máximo de divisiones de comprobación de la escala con inclinómetro	1000 (clase IIII); 5200 (clase III) rango único; 2x5200 o 3x2000 (clase III) intervalo múltiple o rango múltiple
Señal mínima de entrada para división de comprobación de la escala	0.4 µV/VSI
Temperatura de trabajo	-10 °C +40 °C



Utilizar una fuente de alimentación externa 12-24 VDC de tipo LPS o en clase 2.

CARACTERÍSTICAS METROLÓGICAS DE LOS INSTRUMENTOS HOMOLOGADOS

Normas respetadas	2014/31/UE - EN45501:2015 - OIML R76:2006
Modos de funcionamiento	rango único, intervalo múltiple, rango múltiple
Clase de precisión	III o IIII
Número máximo de divisiones de comprobación de la escala	10000 (clase IIII); 1000 (clase IIII)
Número máximo de divisiones de comprobación de la escala con inclinómetro	1000 (clase IIII); 5200 (clase III) rango único; 2x5200 o 3x2000 (clase III) intervalo múltiple o rango múltiple
Señal mínima de entrada para división de comprobación de la escala	0.4 µV/VSI
Temperatura de trabajo	-10 °C +40 °C



3.5. Ficha de datos de la cámara HIKVISION.

Specifications

Camera	
Image Sensor	1/1.8" Progressive Scan CMOS
Min. Illumination	Color: 0.002 Lux @ (F1.2, AGC ON), 0 Lux with IR
Shutter Speed	1 s to 1/100,000 s
Slow Shutter	Yes
Day & Night	IR Cut Filter
Digital Noise Reduction	3D DNR
WDR	140 dB
Angle Adjustment	Bracket, pan: 0° to 355°, tilt: 0° to 90°, rotate: 0° to 360°
Lens	
Focal length	2.8 to 12 mm 8 to 32 mm
Aperture	2.8 to 12 mm: F1.2 8 to 32 mm: F1.6
Focus	Auto, semi-auto, manual
FOV	2.8 to 12 mm: horizontal FOV 103.3° to 38.6°, vertical FOV 54.2° to 21.9°, diagonal FOV 124.2° to 44.3° 8 to 32 mm: horizontal FOV 42.5° to 13.4°, vertical FOV 23.4° to 7.7°, diagonal FOV 49° to 15.3°
Lens Mount	Integrated
IR	
IR Range	2.8 to 12 mm: up to 50 m 8 to 32 mm: up to 100 m
Wavelength	850 nm
Compression Standard	
Video Compression	Main stream: H.265/H.264/H.265+/H.264+ Sub stream/third stream/fourth stream/fifth stream/custom stream: H.265/H.264/MJPEG
H.264 Type	Baseline Profile/Main Profile/High Profile
H.264+	Main stream supports
H.265 Type	Main Profile
H.265+	Main stream supports
Video Bit Rate	32 Kbps to 16 Mbps
Audio Compression	No
Audio Bit Rate	No
Smart Feature-set	
Perimeter Protection	Line crossing detection, intrusion detection, region entrance detection, region exiting detection, unattended baggage detection, object removal detection
Exception Detection	Scene change detection, defocus detection
Statistics	No
Face Detection	Yes
Recognition	License Plate Recognition
Region of Interest	4 fixed regions for main stream, sub stream, third stream, fourth stream, and fifth stream, and dynamic tracking
Road Traffic and Vehicle Detection	

Accuracy (Under recommended installation and lighting conditions)	Capture rate > 98% Vehicle moving direction recognition accuracy > 96% Mistaken capture rate < 2% (entrance/exit), < 5% (checkpoint)
Blacklist and whitelist	Max 10,000 records
No License Plate Detection	Yes
Motorcycle LPR	Yes (only applicable to checkpoint)
Image	
Max. Resolution	1920 × 1080
Main Stream	50Hz: 50fps (1920 × 1080, 1280 × 960, 1280 × 720) 60Hz: 60fps (1920 × 1080, 1280 × 960, 1280 × 720)
Sub Stream	50Hz: 25fps (704 × 576, 640 × 480) 60Hz: 30fps (704 × 480, 640 × 480)
Third Stream	50Hz: 25fps (1920 × 1080, 1280 × 960, 1280 × 720, 704 × 576, 640 × 480) 60Hz: 30fps (1920 × 1080, 1280 × 960, 1280 × 720, 704 × 480, 640 × 480)
Fourth Stream	50Hz: 25fps (1920 × 1080, 1280 × 720, 704 × 576, 640 × 480) 60Hz: 30fps (1920 × 1080, 1280 × 720, 704 × 480, 640 × 480)
Fifth Stream	50Hz: 25fps (704 × 576, 640 × 480) 60Hz: 30fps (704 × 480, 640 × 480)
Custom Stream	50Hz: 25fps (1920 × 1080, 1280 × 720, 704 × 576, 640 × 480) 60Hz: 30fps (1920 × 1080, 1280 × 720, 704 × 480, 640 × 480)
Image Enhancement	BLC, HLC, 3D DNR, Defog, EIS, lens distortion correction
Image Setting	Rotate mode, saturation, brightness, contrast, sharpness, AGC, and white balance are adjustable by client software or web browser
Target Cropping	Yes
SVC	H.264 and H.265 encoding
Day/Night Switch	Day/Night/Auto/Schedule/Triggered by Alarm In/ Triggered by video
Picture Overlay	LOGO picture can be overlaid on video with 128 × 128 24bit bmp format
Network	
Network Storage	microSD/SDHC/SDXC card (256 GB), local storage and NAS (NFS, SMB/CIFS), ANR
Alarm Trigger	Motion detection, video tampering alarm, network disconnected, IP address conflict, illegal login, HDD full, HDD error, alarm for listed license plate in the blacklist and whitelist
Protocols	TCP/IP, ICMP, HTTP, HTTPS, FTP, DHCP, DNS, DDNS, RTP, RTSP, RTCP, PPPoE, NTP, UPnP, SMTP, SNMP, IGMP, 802.1X, QoS, IPv6, UDP, Bonjour
Security Measures	Password protection, HTTPS encryption, IEEE 802.1x port based network access control, IP address filter, basic and digest authentication for HTTP/HTTPS, WSSE and digest authentication for ONVIF
General Function	One key reset, anti flicker, five streams and custom stream, heartbeat, password protection, privacy mask, watermark, IP address filter
API	ONVIF (PROFILE S, PROFILE G), ISAPI, SDK
Simultaneous Live View	Up to 20 channels
User/Host	Up to 32 users. 3 user levels: administrator, operator and user
Client	iVMS 4200, Hik Connect, iVMS 5200, Hik Central
Web Browser	Plug-in required live view: IE8+, Chrome31.0 44, Mozilla Firefox30.0 51, Safari8.0+ Plug-in for live view: Chrome45+, Mozilla Firefox52+
Interface	
Audio	No
Communication Interface	1 RJ45 10M/100M/1000M Ethernet port, 1 RS-485 interface (half duplex, HIKVISION, Pelco-P, Pelco-D, self-adaptive)

Alarm	2 inputs, 2 outputs (up to 24 VDC 1A or 110 VAC 500 mA)
Video Output	1Vp-p composite output (75 Ω /CVBS)
On-board storage	Built-in micro SD/SDHC/SDXC slot, up to 256 GB
BNC	CVBS analog output (4CIF resolution), internal
Reset Button	Yes
Audio	
Environment Noise Filtering	No
Audio Sampling Rate	No
General	
Firmware Version	5.5.60
Operating Conditions	- 30 °C to 60 °C (22 °F to 140 °F), H: 40 °C to 60 °C (40 °F to 140 °F) Humidity: 95% or less (non-condensing)
Power Supply	12 VDC \pm 20%, two core terminal block PoE (802.3at, class 4)
Power Consumption and Current	- IZS: 12 VDC, 1.2 A, max. 14 W; PoE (802.3at, 42.5 V to 57 V), 0.4 A to 0.3 A - IZHS: 12 VDC, 1.4 A, max. 16.5 W; PoE (802.3at, 42.5 V to 57 V), 0.4 A to 0.3 A
Protection Level	IP67, IK10
Heater	- H: yes
Material	Aluminum alloy
Dimensions	Camera: Φ 140 \times 351 mm (Φ 5.5" \times 13.8") With package: 405 \times 190 \times 180 mm (15.9" \times 7.5" \times 7.1")
Weight	Camera: approx. 2.5 kg (5.51 lb.) With package: approx. 2.75 kg (6.06 lb.)

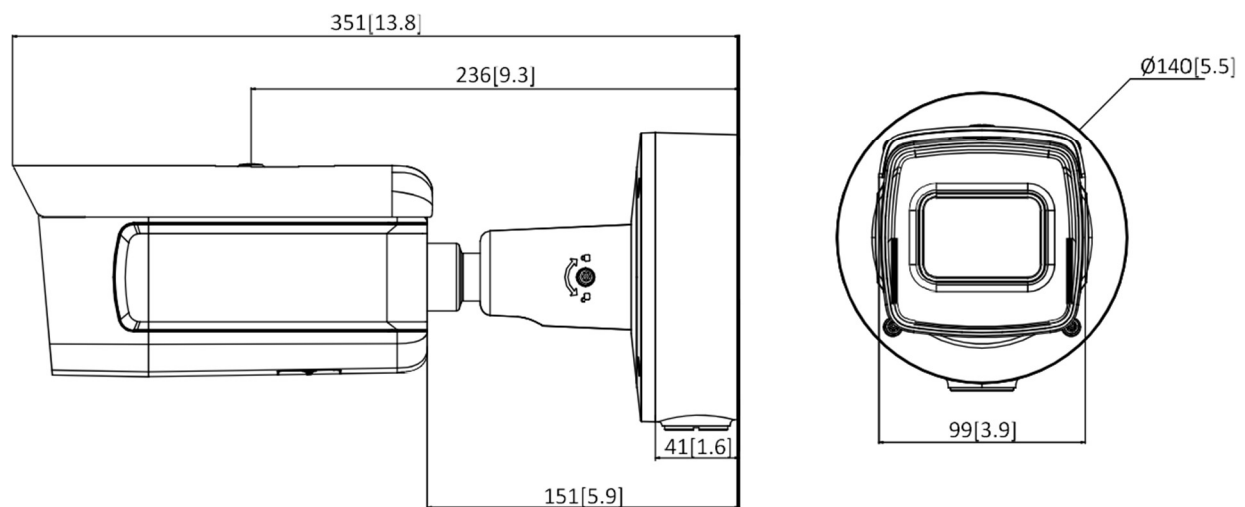
* Listed resolutions are only selectable options. It does not mean that all streams can work at their maximum resolution at the same time.

Available Model

DS 2CD7A26G0/P IZS (2.8 12 mm), DS 2CD7A26G0/P IZHS (2.8 12 mm), DS 2CD7A26G0/P IZS (8 32 mm), DS 2CD7A26G0/P IZHS (8 32 mm)

* H: Heater supported

Dimensión



Unit: mm[inch]

Accessory



DS 1475ZJ SUS
Vertical Pole Mount



DS 1476ZJ SUS
Corner Mount



DS 2251ZJ/PM LPR
Column Mount

3.6. Ficha de datos de la placa TRAFFIC VF de la barrera automática.

ES

Central de mando

1. INTRODUCCIÓN

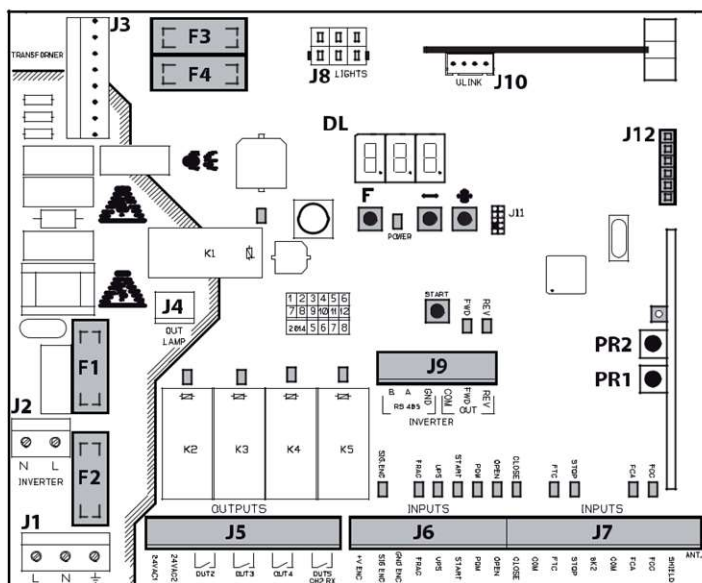
 La central de mando ha sido diseñada para controlar barreras automáticas.



= Conexión eléctrica ya predisuelta en fábrica.

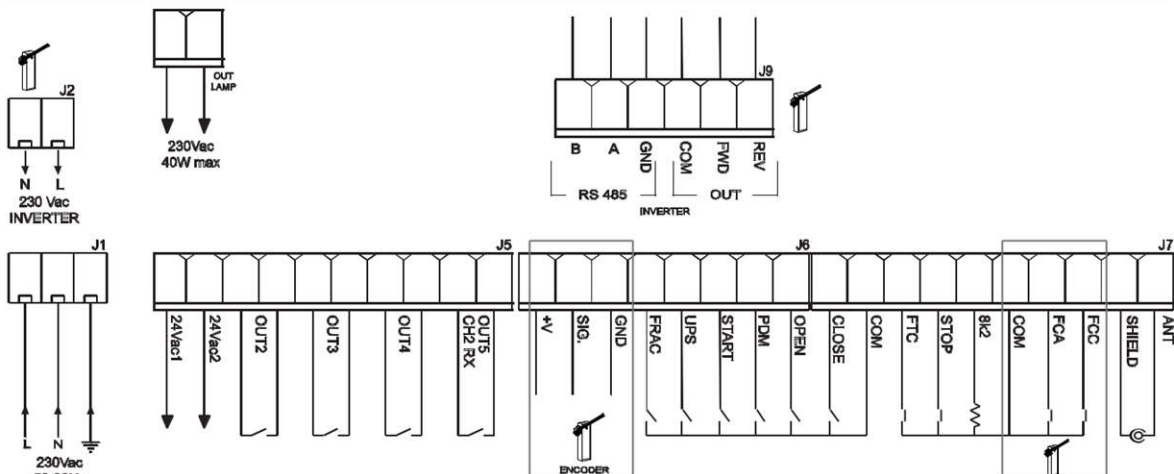
2. CARACTERÍSTICAS PRINCIPALES

- Lógica de control por microprocesador
- Pilotos que muestran el estado de las entradas
- Módulo TCP/IP y RS485 (Opcional)
- Pantalla 3 dígitos para el estado del sistema y programación
- Hasta 4 salidas configurables
- Calefactor incorporado para climas fríos (Termon)



- J1: Alimentación central de mando
J2: Alimentación inverter
J3: Conector Transformador
J4: Salida indicador parpadeante
J5: Alimentación accesorios/salida
J6: Encoder/entradas
J7: Entradas/antena
J8: Conector alimentador de luces
J9: Señales inverter

- J10: Conector expansión
DL: Pantalla 3 dígitos
START: Tecla de mando "START"
F1: Fusible transformador: 500 mA T (230 Vac) - 5x20 mm
F2: Fusible de línea: 4 AT (230 Vac) - 5x20 mm
F3, F4: Fusible baja tensión: 2 AT 5x20 mm
F, -, +: Pulsadores de programación



ESPAÑOL

3. CARACTERÍSTICAS TÉCNICAS

- Alimentación:.....230Vac \pm 10%, 50/60Hz
- Salida luz intermitente:.....230Vac; 40W max
- Salida accesorios:.....24Vac; 1A max

4. SEGURIDAD DE LA INSTALACIÓN

Para que se alcance el grado de seguridad requerido por la normativa vigente, lean atentamente las siguientes prescripciones.

- 1) Realicen todas las conexiones en el tablero de bornes leyendo atentamente las indicaciones incluidas en este manual y respetando las normas generales y de buena técnica que regulan la ejecución de las instalaciones eléctricas.
- 2) Preparar antes de la instalación un interruptor magnetotérmico omnipolar con una distancia de apertura de los contactos de un mínimo de 3 mm.
- 3) Instalar, si no está previsto, un interruptor diferencial con umbral 30 mA.
- 4) Comprobar la eficacia de la instalación de toma de tierra y conectar a ésta todas las partes del automatismo provistas de borne o cable de tierra.
- 5) Prever la presencia de al menos un dispositivo de señalación exterior, de tipo semáforo o luz intermitente, acompañado de un cartel de indicación de peligro o de aviso.
- 6) Aplicar todos los dispositivos de seguridad requeridos por el tipo de instalación considerando los riesgos que ésta puede causar.
- 7) Separar en las canalizaciones las líneas de potencia (1,5 mm² tamaño mínimo) de las de señal de baja tensión (0,5 mm² tamaño mínimo).



5. OPERACIONES PRELIMINARES

Antes de dar un mando al automatismo, comprobar que haya sido seleccionado correctamente el tipo de barrera/mástil en el siguiente modo: - Para seleccionar la barrera conectada, mantener pulsadas las teclas F y + durante 5 segundos.

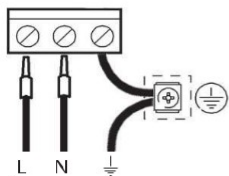
- Seleccionar el tipo de barrera/mástil utilizando los pulsadores +/-.
- Para confirmar pulsar las teclas F y +.
- Seleccionar exclusivamente la barrera/mástil utilizado:

SELECCIÓN DEL TIPO DE BARRERA			
E-B	Park 60 Plus XT	90	Mástil articulado H65 (sólo Park Plus 30 XT)
3-5	Park 30 Plus XT	180	No utilizado
Erb	Mástil Carbon (sólo Park 30 Plus XT)		

La Empresa no es responsable por los daños causados a personas, animales o bienes a causa de una selección incorrecta del tipo de barrera/mástil. La selección incorrecta del tipo de barrera/mástil hará caducar la garantía.

6. CONEXIONES Y FUNCIONES DE ENTRADAS Y SALIDAS

6.1 J1 ALIMENTACIÓN CENTRAL DE MANDO

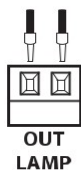


Alimentación a 230V 50/60 Hz.

Conectar la fase y el neutro como se muestra en la serigrafía. Utilizar un cable de tipo H07RN-F 2x1,5+T min. Conecte el conductor amarillo/verde de la red de alimentación al borne de tierra del aparato.

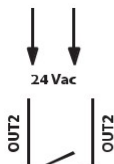


6.2 J4 SALIDA PARPADEANTE



Salida 230Vac, 40W max.

6.3 J5 ALIMENTACIÓN ACCESORIOS/SALIDAS



OUT24

Salida 24Vac, 1A MAX



OUT2

Salida programable de relé con contacto limpio 500mA máx., 24Vca/cc (parámetro a2 - 2º nivel)



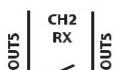
OUT3

Salida programable de relé con contacto limpio 500mA máx., 24Vca/cc (parámetro a3 - 2º nivel)



OUT4

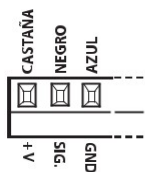
Salida programable de relé con contacto limpio 500mA máx., 24Vca/cc (parámetro a4 - 2º nivel)



OUT5/CH2 RX

Salida programable N.A. del 2º canal receptor radio integrado (parámetro a5 - 2º nivel)

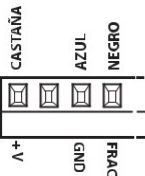
6.4 J6 ENCODER/ENTRADAS

**ENCODER**

Se entrega cableado y probado. El dispositivo interviene solo en la fase de cierre cuando el mástil golpea un obstáculo.

Seleccionar el programa deseado mediante la programación del parámetro EC-1ºniv.

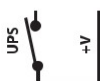
El terminal +V puede utilizarse para alimentar los sensores suplementarios (16 Vdc no estabilizada-100 mA máx.)

**FRAC**

Entrada adicional N.C. de seguridad.

Es posible conectar el sensor del mástil abatible para impactos

Cuando se activa (abierto) detiene de inmediato el automatismo y un arranque sucesivo provoca siempre una apertura.

**UPS**

Entrada estado UPS/red de alimentación.

Para utilizar con UPS con salida de señalización específica (contacto cerrado en caso de UPS activo)

La central también tiene un sistema interior de detección de la forma de onda que no requiere el uso de esta entrada con sistemas UPS en forma de onda cuadrada.

**START**

Entrada N.O. que permite mandar la automatización según la lógica abre, stop, cierra, abre.

**PDM**

Entrada programable Pd-3er nivel.

Es posible replicar la señal en una salida configurada a fin de tener un contacto de potencia

**OPEN**

Entrada N.O. sólo de abertura.

Manteniendo controlada esta entrada el automatismo efectuará la maniobra de apertura y efectuará el eventual reenganche automático sólo cuando se haya liberado la entrada. Conectar aquí eventuales relojes o timer diarios o semanales.

6.5 J7 ENTRADAS/ANTENA

**CLOSE**

Entrada N.O. de cierre. Permite cerrar el automatismo sólo si los dispositivos de seguridad no están ocupados.

Modalidad de funcionamiento programable mediante el parámetro CL-1er nivel.

**FTC**

Entrada N.C. de seguridad (fotocélula). Ingresar el programa deseado mediante la programación del parámetro FE-1er nivel. El dispositivo interviene solo en la fase de cierre, nunca interviene en la fase de apertura.

**STOP**

Entrada N.C. de seguridad. Cuando se activa detiene inmediatamente la automatización y un sucesivo start provoca siempre una re-apertura. Durante el tiempo de pausa (trimmer PAUSE) un mando de Stop elimina el re-cierre automático dejando la barrera abierta a la espera de mandos.

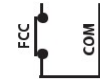
N.B.: Con esta entrada ya está conectado de serie el microinterruptor de la puerta

**8k2**

Entrada analógica para funciones múltiples. Para TERMON véase el apartado 8.3.

**FCA**

Entrada N.C. de final de carrera en apertura. Cuando se activa termina la carrera de apertura.

**FCC**

Entrada N.C. de final de carrera en cierre. Cuando se activa termina la carrera de cierre.

**ANTENNA**

Conexión antena para receptor integrado

9.2 PROGRAMACIÓN DE 1º NIVEL

En el cuadro siguiente se presentan las funciones de 1º nivel y los parámetros individuales configurables.



= valor por DEFECTO configurado en fábrica.



= valor del parámetro configurado en la fase de instalación: se debe indicar en caso de que se modifique el valor por DEFECTO.

Par	Función	Rango de valores a establecer		
L0	Selecciona la lógica de funcionamiento. (ver notas después del cuadro)	00: Hombre presente	01	
		01: Semiautomático		
		02: Automático		
CL	Configuración entrada Close (ver notas después del cuadro)	00: Entrada Close estándar	00	
		01: Entrada Close de liberación		
		02: El comando cerrar funciona de cierre a liberación y seguridad		
Ft	Fotocélulas	00: En cierre para y espera mandos con fotocélula libre	02	
		01: En cierre para; cierra al cabo de 1" con fotocélula libre		
		02: En cierre vuelve a abrir; cierra al cabo de 1" con fotocélula libre		
		03: En cierre vuelve a abrir; cierra al cabo de 5" con fotocélula libre		
		04: En cierre vuelve a abrir; cierra al librarse la fotocélula		
		05: En cierre vuelve a abrir y espera mandos con fotocélula libre.		
EC	Encoder	00: Excluido	03	
		01: En cierre para y espera mandos		
		02: En cierre vuelve a abrir y espera mandos		
		03: En cierre vuelve a abrir, cierra al cabo de 5 segundos		
ES	Sensibilidad Encoder	01-09 (menor - mayor)	01	
PF	Parpadeo previo	00: Excluido	00	
		01: Antes de cada movimiento en una salida configurada (ver Parámetros 02, 03, 04, 05 en el cuadro nivel 2º)		
		02: Antes de cada movimiento en una salida configurada y en las luces barra		
Lb	Luces barra	00: Las luces rojas parpadean durante el movimiento, las luces rojas se apagan con el mástil cerrado y abierto	03	
		01: Las luces rojas parpadean durante el movimiento, las luces rojas se encienden con el mástil cerrado y se apagan con el mástil abierto		
		02: Las luces rojas parpadean durante el movimiento y con el mástil cerrado, las luces rojas se encienden durante la parada y las luces rojas se encienden con el mástil abierto		
		03: Las luces rojas parpadean durante el movimiento, las luces rojas se apagan con el mástil cerrado y las luces verdes se encienden con el mástil abierto		
		04: Las luces rojas parpadean durante el movimiento, las luces rojas se encienden con el mástil cerrado y las luces verdes se encienden con el mástil abierto		
EP	Tiempo de pausa (expresado en segundos)	00-99	10	
dF	Restablecimiento parámetros por defecto. (ver notas después del cuadro)	00: Ningún restablecimiento	00	
		01: Restablecimiento parámetros por defecto		
		02: Restauración de parámetros por defecto y tipo de barreras excepto el parámetro «Com»: protocolo de comunicación		
St	Salida menú/salvamento	Pulsando la tecla «F» se sale del modo de programación y se memorizan las modificaciones realizadas		

Descripción parámetros nivel 1

· L0: Lógica de funcionamiento

- Hombre presente: La automatización funciona para mandos mantenidos. El comando de start una vez abre y una vez cierra.
- Semiautomática: La automatización funciona para mandos de impulsos sin el cierre automático. Por lo tanto, al final de la apertura para mandar el cierre hay que actuar respectivamente sobre el start o sobre el close.
- Automática: La automatización funciona por impulsos. En el ciclo normal, terminada la fase de apertura es activado el cierre automático una vez transcurrido el tiempo de pausa programado (parámetro EP).

• **CL: Configuración Close**

- 01: Entrada Close de liberación

Modalidad de funcionamiento estudiada para tener el cierre automático de la barra cuando el vehículo ha dejado libre la fotocélula o del detector magnético (accesorios más idóneos para esta utilización). Conectar el contacto N.O. del detector o de la fotocélula a los bormes del contacto Close. La presencia del vehículo en el detector o delante de la fotocélula no provoca el cierre inmediato sino que hay que esperar la liberación de la señal correspondiente.

- 02: El comando cerrar funciona de cierre a liberación y seguridad.

Durante la fase de cierre la activación del comando cerrar detiene la automatización. A la desactivación la barrera reanuda el cierre.

• **dF: Default**

- Para restaurar los parámetros por defecto es necesario configurar en 1 o 2 el parámetro dF y salir del menú. Con 2 se preserva el ajuste correspondiente a la comunicación (Com)

9.3 PROGRAMACIÓN DE 2º NIVEL

En el cuadro siguiente se presentan las funciones de 2º nivel y los parámetros individuales configurables.



= valor por DEFECTO configurado en fábrica.



= valor del parámetro configurado en la fase de instalación: se debe indicar en caso de que se modifique el valor por DEFECTO.

Par	Función	Rango de valores a establecer		
CL	Tiempo de trabajo máximo (seg.)	03-30	15	
5r	Solicitud de mantenimiento	00: Inhabilitada 01: activa en las salidas configuradas 02: activa grupo salidas configuradas y doble parpadeo en luces barra	00	
nt	Programación ciclos de mantenimiento en miles	00-99	00	
nL	Programación ciclos de mantenimiento en millones	0.0-9.9	0.0	
02 03 04 05	Output 2, Output 3, Output 4, Output 5	00: solicitud de mantenimiento 01: actuación fotocélula 02: actuación reverser 03: contacto PDM activado 04: barra cerrada 05: barra abierta 06: contacto stop activado 07: parpadeo previo 08: bloqueo barra 09: contacto open activo 10: contacto close activo 11: contacto start activo 12: contacto FRAC 13: contacto UPS 14: contacto segundo canal radio (solo para OUT 5)	14 50 20 10 40 60 20 50	
EE	Termon	00: deshabilitado 01: habilitado siempre activo 02: habilitado con sensor de temperatura NTC conectado entre 8k2 y COM	00	
UP	UPS	00: deshabilitado 01: habilitado, apertura automática en caso de fallo de corriente eléctrica 02: habilitado, cierre automático en caso de fallo de corriente eléctrica ⚠ ATENCIÓN: SELECCIÓN PELIGROSA	00	
5t	Salida menú/guardado	Pulsando la tecla «F» se sale del modo de programación y se memorizan las modificaciones realizadas		

Descripción parámetros nivel 2**• $\overline{S}r$: Solicitud de mantenimiento**

$\overline{00}$: la solicitud de mantenimiento no se activa.

$\overline{01}$: al final de la cuenta atrás, efectuada por medio de los contadores nL y nL , es activada una de las salidas programadas (véase parámetro $\overline{02}, \overline{03}, \overline{04}, \overline{05}$)

$\overline{02}$: al final de la cuenta atrás, efectuada por medio de los contadores nL y nL , es activada una de las salidas programadas (véase parámetro $\overline{02}, \overline{03}, \overline{04}, \overline{05}$) y las luces de la barra efectúan un doble parpadeo.

• nL ed nL : Programación ciclos de mantenimiento en miles y millones

La combinación de los dos parámetros permite configurar una cuenta atrás después de la cual es señalada la solicitud de mantenimiento.

El parámetro nL permite configurar los miles, el parámetro nL los millones.

Ejemplo: para configurar 275.000 maniobras de mantenimiento hay que programar nL en 0.2 y nL en 75.

El valor visualizado en los parámetros se actualiza con la sucesión de las maniobras.

• $\overline{E}E$: THERMON (sistema electrónico integrado de calentamiento del motor).

$\overline{01}$: sistema está siempre activo. Siempre debe ser utilizado solo con temperatura ambiente máxima $\leq 10^{\circ}\text{C}$.

⚠ ATENCIÓN: en caso de temperatura ambiente $> 10^{\circ}\text{C}$ se puede producir un recalentamiento y averiarse el motor, lo que no está amparado por la garantía

$\overline{02}$: el sistema se activa según la temperatura del motor medida con la sonda NTC (opcional). En caso de avería de la sonda, el sistema restablece la configuración $\overline{01}$.

• \overline{UP} : UPS

TIPO UPS	
UPS de onda cuadrada	Detector interior, entrada UPS no conectada
UPS de onda sinusoidal pura o casi sinusoidal con salida señalización de ausencia de red	Conectar la señal de ausencia de red en la entrada UPS
UPS de onda senoidal pura sin salida de señalización de avería	Utilizar un relé de 230 Vac con bobina conectada a la red y contactos en la entrada UPS

• Configuración bloqueo barra:

Para utilizar el bloqueo de barra conecte el contacto de habilitación con OUT2 o bien OUT3 o bien OUT4 o bien OUT5 y configure en $\overline{08}$ el parámetro correspondiente $\overline{02}, \overline{03}, \overline{04}$ o bien $\overline{05}$. Configure avance desactivación bloqueo ($r5 - 3^{\circ}$ nivel).

9.4 PROGRAMACIÓN DE 3° NIVEL

En el cuadro siguiente se presentan las funciones de 3° nivel y los parámetros individuales configurables.



= valor por DEFECTO configurado en fábrica.



= valor del parámetro configurado en la fase de instalación: se debe indicar en caso de que se modifique el valor por DEFECTO.

Par	Función	Rango de valores a establecer		
\overline{AS}	Advanced setup	$\overline{00}$: ningún setup avanzado	$\overline{00}$	
		$\overline{01}$: N/A		
		$\overline{02}$: controlada, y salida automática		
\overline{Pd}	Polaridad de entrada dinámica PDM	$\overline{00}$: entrada N.O.	$\overline{00}$	
		$\overline{01}$: entrada N.C.		
$\overline{P2}$ $\overline{P3}$ $\overline{P4}$ $\overline{P5}$	Polaridad salida 2 Polaridad salida 3 Polaridad salida 4 Polaridad salida 5	$\overline{00}$: N.O. $\overline{01}$: N.C.	$\overline{00}$	
$\overline{r5}$	Avance desactivación bloqueo	$\overline{00}$: 0,5s - $\overline{01}$: 1s - $\overline{02}$: 1,5s - $\overline{03}$: 2s - $\overline{04}$: 2,5s - $\overline{05}$: 3s	$\overline{00}$	
$\overline{05}$	Salida menú/salvamento	$\overline{66} - \overline{99}$ (Xtreme 68, BL-SMALL-SN 90°, BL-SMALL-SN 180°)	$\overline{66}$	
		$\overline{50} - \overline{99}$ (Xtreme 35, Carbon)	$\overline{50}$	
\overline{CS}	Velocidad de cierre (%)	$\overline{53} - \overline{99}$ (Xtreme 68, BL-SMALL-SN 90°, BL-SMALL-SN 180°)	$\overline{53}$	
		$\overline{40} - \overline{99}$ (Xtreme 35, Carbon)	$\overline{40}$	
\overline{FP}	Entrada selección velocidad	$\overline{00}$: Inhabilitada	$\overline{00}$	
		$\overline{01}$: Habilitada		
\overline{Fr}	Sensor de entrada del mástil abatible para impactos	$\overline{00}$: sensor del mástil abatible para impactos no montado o deshabilitado	$\overline{00}$	
		$\overline{01}$: sensor del mástil abatible para impactos montado y activo N.C.		
\overline{St}	Salida menú/salvamento	Pulsando la tecla «F» se sale del modo de programación y se memorizan las modificaciones realizadas		

3.7. Ficha de datos del detector de lazo inductivo.

2.1 Características principales

- Inductancia de entrada del lazo: 20 μ H a 1.000 μ H
- Sensibilidad ajustable en 10 pasos: 0,01 % a 1,00 % con potenciómetro
- Sintonización automática o manual de la frecuencia del lazo a través de 4 canales de frecuencia del lazo ajustables para evitar las interferencias
- Aumento automático de la sensibilidad (ASB) para la detección de vehículos con plataforma alta
- Selección entre modo de fallo y modo seguro
- 2 salidas SPDT seleccionables para pulso y presencia
- Indicación LED multicolor de alimentación/fallo para una configuración sencilla y un diagnóstico intuitivo
- LED multicolor individual de estado del lazo para indicar los distintos estados del lazo y del fallo
- Capacidad de diagnóstico del lazo: cortocircuito del lazo, circuito abierto del lazo, inductancia fuera de rango, interferencias entre canales
- Lógica direccional para doble lazo
- Amplio rango de alimentación: 24-240 CA/CC, 45-65 Hz

2.2 Número de identificación

Código	Opción	Descripción
L	-	Lazo
D	-	Detector
D	-	Carril DIN
1/2	1	Número de lazos
	2	Número de lazos
P	-	Potenciómetro
A	-	Ajuste
2	-	Número de salidas
D	-	2 salidas SPDT
U24	-	Alimentación 24-240 VCA/VCC

Configuración de las patillas para un lazo (LDD1)

L11	Lazo
L12	Lazo
E1	Tierra
R21	Relé 2 normalmente abierto (NA)
R22	Relé 2 normalmente cerrado (NC)
R23	Relé 2 común (COM)
R11	Relé 1 normalmente abierto (NA)
R12	Relé 1 normalmente cerrado (NC)
R13	Relé 1 común (COM)
A1	Alimentación
A2	Alimentación
E2	No conectado



2.3 Especificaciones

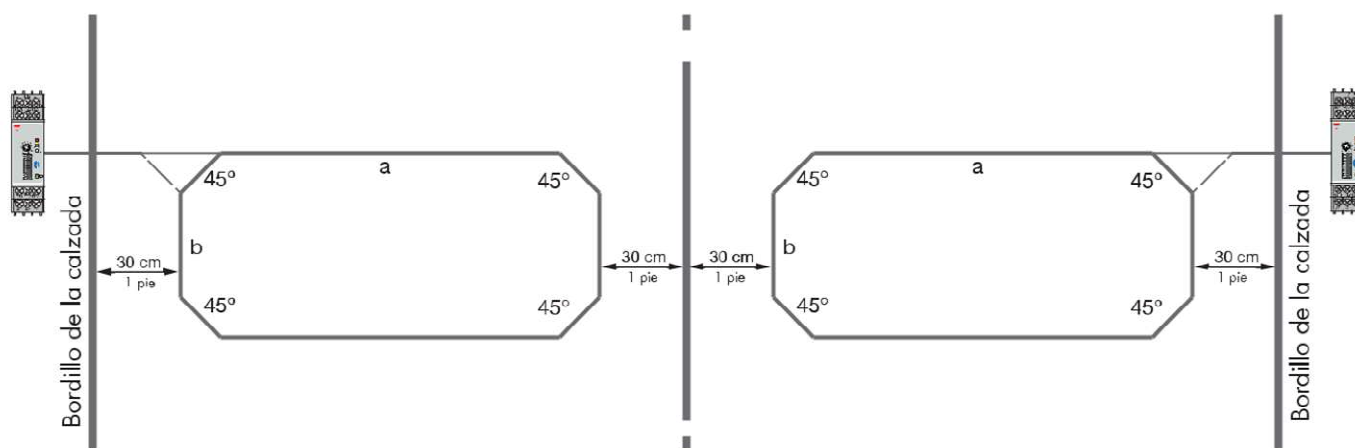
Inductancia de entrada del lazo	20 μ H ... 1000 μ H
Sensibilidad ajustable	0,01% ... 1,00%
Número de pasos ajustables	10
Número de canales de frecuencia	4
Rango de frecuencia	10 ... 130 kHz
Detección de fallos del lazo	Cortocircuito, circuito abierto, inductancia fuera de rango, interferencias en la frecuencia
Tiempo de respuesta	130 ms
Tipo de salida	Relé
Número de salidas	2 x SPDT
Modo de salida	Pulso o presencia; seleccionable a través de interruptores DIP
Asignación de salida	LDD1: 2 SPDT para lazo 1 LDD2: 1 SPDT para lazo 1 y 1 SPDT para lazo 2
Tensión operación máx. relé	250CA/CC
Intensidad nominal de funcionamiento (I_n) relé	CA1: 5A@250 VCA CC1: 1A@30 VCC
Vida mecánica	15 x 10 ⁶
Vida eléctrica	>100.000 operaciones (a 5 A de carga)
Protección	Inversión de polaridad, sobretensión
Rango de tensión de alimentación (U_e)	24 ... 240 VCA/VCC
Consumo LDD1	24 VCA/VCC < 2 W / 2,5 VA 115 VCA/VCC < 2 W / 3 VA 240 VCA/VCC < 2 W / 4 VA
Frecuencia de alimentación nominal	45 ... 65 Hz
Tensión nominal de aislamiento	800 V
Pulso de tensión soportada	4 kV (1,2/50 μ s)
Retardo a la conexión (t_v)	< 5 s con sintonización manual de los canales de frecuencia < 10 s con sintonización automática de los canales de frecuencia
Temperatura ambiente	-40° ... +70°C (-40° ... +158°F) (De funcionamiento) -40° ... +70°C (-40° ... +158°F) (De almacenamiento)
Rango de humedad ambiental	0% ... 90% (De funcionamiento) 0% ... 90% (De almacenamiento)
Categoría de sobretensión	III (IEC)
Grado de protección	IP20 (IEC)
Grado de contaminación	2 (IEC)
Tipo de conexión	Terminal a tornillo
Material de la caja	PPO PX9406-802, PPO Noryl SE1
Color	RAL 7035 (Gris)
Tamaño	84 x 22 x 99 mm (Al x An x P)
Peso	LDD1: 134 g LDD2: 139 g

7. Instalación del lazo

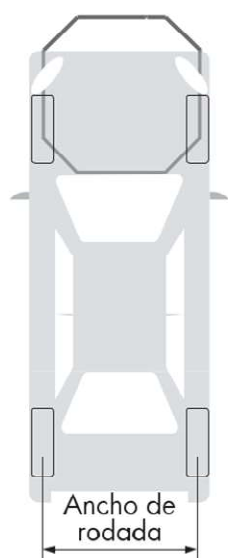
Una instalación adecuada del lazo en la calzada es el único y más importante factor para obtener un sistema de detección fiable. La mayoría de problemas de detección están provocados por una instalación incorrecta del lazo. Lea con atención las siguientes directrices a fin de garantizar el mejor rendimiento de la aplicación.

7.1. Dimensiones y colocación del lazo

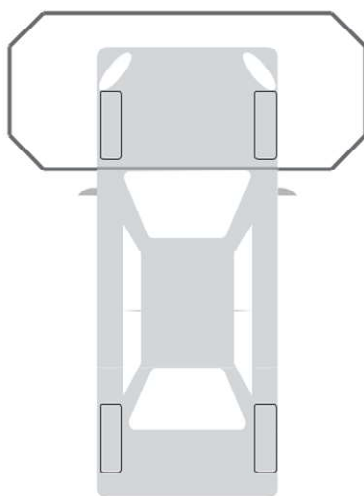
Lo primero que debe tenerse en cuenta para la instalación de un lazo nuevo son las dimensiones y la colocación. Las dimensiones del lazo dependen del tamaño de la calzada y normalmente su forma es rectangular con los bordes en chaflán. El lazo debe colocarse con una distancia de aproximadamente 30 cm respecto al borde de la calzada y al resto de carriles. Esto ayuda a prevenir falsas detecciones causadas por el tráfico de los carriles adyacentes.



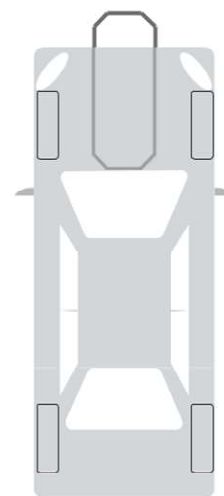
Para reducir la tensión en el cable y prolongar así su vida útil, es importante evitar las curvaturas bruscas del cable. Para ello, es necesario realizar cortes con ángulos de 45 grados en las esquinas de la forma rectangular. El mejor estado de la señal para la detección de un vehículo se obtiene cuando la anchura del lazo (a) es más o menos igual que el ancho de rodada del vehículo. Por eso, lo ideal es que la anchura del lazo sea igual que la anchura de los vehículos que deban detectarse.



Favorable

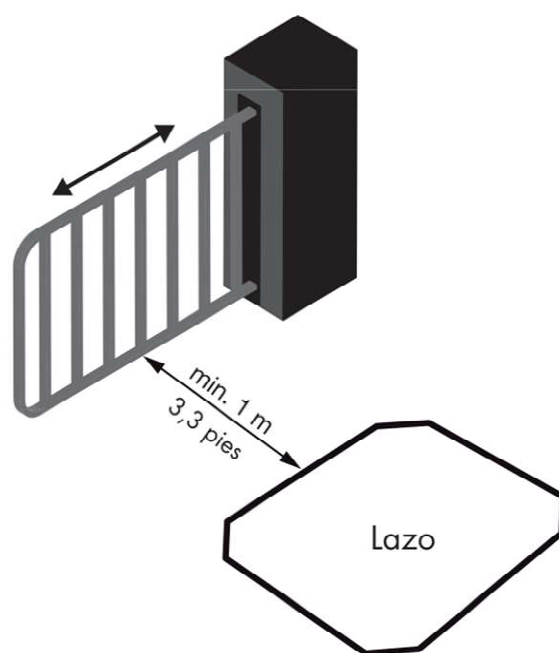


Desfavorable



Desfavorable

Cuando el detector de lazo se enciende o se reinicia, este se sintoniza automáticamente con el entorno que lo rodea. Esto significa que los objetos metálicos estáticos, como postes, armarios o vallas, no afectan al funcionamiento del detector de lazo. Sin embargo, es importante garantizar una distancia segura respecto a los objetos metálicos móviles, como las verjas. En aplicaciones en las que hay objetos metálicos móviles, es importante garantizar una distancia mínima de 1 metro entre el lazo y el objeto. De lo contrario, el lazo se vería afectado y ello provocaría falsas detecciones.



La longitud del lazo (b) influye en la velocidad máxima a la que un vehículo puede circular para seguir siendo detectado. Para aplicaciones en las que se requiera la detección de vehículos a alta velocidad es importante tener en cuenta esta longitud. En la tabla de abajo se muestra la relación entre la longitud del lazo (b) y la velocidad máxima del vehículo. En la tabla se da por sentado el ajuste correcto de la sensibilidad del detector de lazo, para una longitud mínima del vehículo de 2,5 metros.

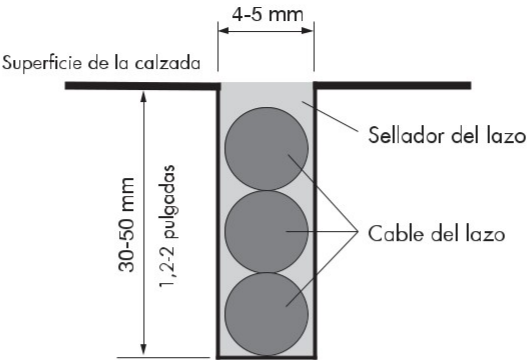
Longitud mínima del lazo (b)	Velocidad máxima del vehículo
0,25 metros	75 km/h
0,50 metros	80 km/h
1,00 metros	95 km/h
2,00 metros	120 km/h
5,00 metros	200 km/h

7.2. Inductancia y giros del lazo

Para garantizar un funcionamiento óptimo, la inductancia del lazo debe ser de entre 80 μH y 300 μH . Para conseguirlo, es necesario ajustar el número de vueltas recomendado en el lazo con respecto a la circunferencia del lazo, de acuerdo con la siguiente tabla.

Vueltas recomendadas (80 μH)	Vueltas mínimas (20 μH)
13	9
7	5
6	4
5	3
4	3
3	2
2	1

Para realizar varias vueltas en el lazo, se recomienda tender los cables como se muestra en la figura de abajo.



La profundidad de la roza recomendada es de 30-50 mm. Si los cables se instalan con una profundidad superior a 50 mm, se reducirá la señal de detección del detector de lazo y esto puede afectar negativamente a la detección de los vehículos de plataforma alta.

⚠ Nota: Un problema habitual en los fallos del lazo es el empalme de cables. Se recomienda utilizar un solo cable continuo sin empalmes. En caso de utilizar empalmes, es necesario soldar los cables. No se permite el uso de terminales roscados o de muelle. Todos los empalmes de cables deben aislarse contra la humedad con tubo termorretráctil con revestimiento adhesivo o equivalente.

7.3. Material del cable del lazo

Es importante seleccionar el tipo de cable adecuado para el lazo. Si el material aislante no es adecuado para la aplicación, la cubierta del cable puede agrietarse o absorber la humedad. La penetración de humedad en la cubierta del cable es un problema habitual que puede provocar un cortocircuito del cable a tierra. Esto puede dar lugar a situaciones en las que la aplicación funcione correctamente si no hay humedad, pero genera fallos con gran humedad o lluvia. Las grietas en el aislamiento del cable pueden provocar problemas similares.

Recomendaciones para el cable:

- Se recomienda utilizar polietileno reticulado (XLPE) como material de aislamiento de cables tanto para el sellado en frío como en caliente.
- El material de aislamiento de PVC (TFFN, THHN, THWN) solo se recomienda para el sellado en caliente y si los cables se encapsulan completamente. De lo contrario, el uso de material de aislamiento de PVC no es recomendable.
- Es importante evitar huecos en el sellado alrededor del cable. Esto puede provocar una acumulación de humedad y causar fallos del lazo.

Para solucionar problemas de cables rotos, se puede utilizar un medidor de aislamiento (mínimo de 500 MΩ). Tira un cable desde el medidor hasta el lazo del cable desconectado y coloque el otro cable del medidor en el suelo. La comprobación debe realizarse con tensión CA.

Resistencia medida	Conclusión
100 a 1.000 MΩ	El lazo está en buen estado
50 a 100 MΩ	La integridad del lazo es cuestionable
0 a 50 MΩ	Es necesario sustituir el lazo

7.4. Cable desde el lazo al detector de lazo

Es importante prestar atención a la instalación del lazo entre el detector de lazo y el lazo. La roza entre la esquina del lazo, hasta el borde de la calzada, debe respetar las mismas recomendaciones que para la instalación del lazo.

⚠ Nota: El cable debe trenzarse al menos con 20 vueltas por metro desde la esquina del lazo, hasta el detector de lazo, y debe fijarse hasta los terminales del detector de lazo.

La longitud máxima recomendada para el cable depende de la sección del cable. Para largas longitudes de cable, la sección del cable debe ser más grande.

Sección del cable [mm ²]
0,75 mm ²
1,50 mm ²
2,50 mm ²

A fin de garantizar una detección fiable, es necesario respetar las normas siguientes:

- El cable desde el lazo al detector de lazo no debe tenderse en paralelo a otros cables eléctricos. Debe establecerse una distancia mínima de 10 cm entre este cable y otros cables.
- Es necesario cortar el exceso de cable para ajustarlo a la longitud adecuada. Nunca se debe enrollar ni apilar dentro del armario eléctrico.
- El cable debe fijarse completamente desde la esquina del lazo hasta el detector de lazo. El movimiento del cable durante el funcionamiento puede ocasionar falsas detecciones.
- Los cables de detectores de lazo contiguos no deben colocarse unos cerca de los otros.

7.5 Instalación en suelo

El cable del lazo puede instalarse en la mayoría de superficies de calzada, pero es importante garantizar una base estable. Las instalaciones en asfalto u hormigón son las más habituales, y las que proporcionan un rendimiento más estable. Es importante que el cable del lazo no se mueva cuando la capa superficial esté sometida a la presión de los vehículos. De lo contrario, es posible que el detector de lazo cree falsas detecciones. La instalación estable del cable es especialmente crítica si el detector de lazo funciona con ajustes de alta sensibilidad o con el ASB activado. El movimiento del cable puede derivar, por ejemplo, de las siguientes situaciones:

- Si la capa superficial es demasiado fina para soportar la carga del vehículo
- Si hay varias rozas repartidas por la superficie
- Si la base situada debajo de la capa superficial no es estable (p. ej., tierra, arena o gravilla sin comprimir)

El lazo se puede instalar en hormigón armado siempre que el lazo se coloque sobre las barras de hierro. Si se requiere calefacción eléctrica para la superficie de la calzada, se recomienda utilizar cables de 2 hilos.

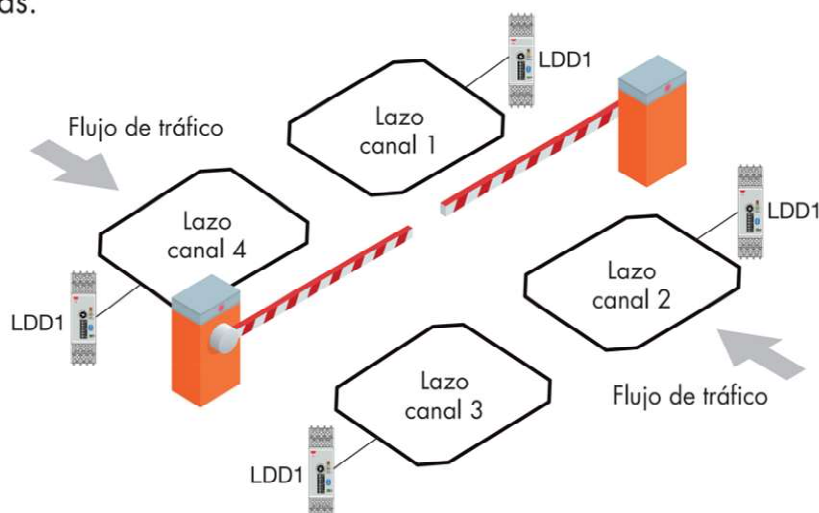
8. Guía de configuración del producto

En el siguiente apartado se expone una introducción general sobre cómo ajustar el detector de lazo. Tenga en cuenta que los cambios realizados en los interruptores DIP no tendrán efecto hasta que no se haya pulsado el botón de reinicio. Es posible realizar ajustes con los potenciómetros de sensibilidad con el equipo en funcionamiento, y los ajustes tendrán efecto inmediatamente sin reiniciar el equipo.

⚠ Advertencia: Antes de realizar cambios en los ajustes del equipo, asegúrese de que los mecanismos de cierre/apertura conectados a la salida del detector de lazo no están a la altura de ninguna persona o vehículo.

8.1 Selección de canales

El detector de lazo puede funcionar en cuatro canales de frecuencia distintos. Esto permite el funcionamiento de hasta cuatro lazos independientes situados cerca sin que influyan unos en los otros. Si hay dos detectores de lazo en funcionamiento en el mismo canal de frecuencia, pueden provocarse interferencias mutuamente y ocasionar falsas detecciones si los lazos están ubicados demasiado cerca uno del otro. Cambiando el canal de frecuencia de uno de los detectores se elimina este problema. En el caso de los detectores de lazo con dos lazos (LDD2), ambos lazos funcionarán en el mismo canal pero no habrá interferencias.



3.8. Ficha de datos de la fotocélula réflex polarizada.

Hoja de características del producto

Características

XUK9APANM12

sensor fotoeléctrico - XUK - polarizado - Sn 5 m -
12..24 VCC - M12

**Principal**

Gama de producto	OsiSense XU
Nombre de serie	Modo único de función genérica
Tipo de sensores electrónico	Sensores fotoeléctrico
Nombre de detector	XUK
Diseño del detector	Compacto 50 x 50
Sistema de detección	Reflexo polarizado
Material	Plástico
Tipo de señal de salida	Discreta
Tipo de circuito de alimentación	DC
Técnica de cableado de detector	3 hilos
Tipo de salida digital	PNP
Función de salida digital	1 NA
Consecutivo, seguido, continuo, adosado	1 conector macho M12, 4 patillas
Aplicación específica de producto	-
Emisión	Rojo reflexo polarizado
Distancia de detección nominal	6 m reflexo polarizado es necesario o reflector XUZY50

Complementario

Material del envoltorio	PBT
Material de la lente	PMMA
Distancia de detección máxima	8 m reflexo polarizado
Tipo de salida	Estado sólido
Material aislamiento cables	PVC
LED de estado	1 LED (amarillo) para código de fecha de fabricación estado de salida
[Us] Tensión nominal de alimentación	12...24 V DC con capacidad de sujeción: protección de polaridad inversa
Límites de tensión de alimentación	10...36 V
Capacidad de conmutación en mA	<= 100 mA (protección contra sobrecargas y cortos-circuitos)
Frecuencia de conmutación	<= 250 Hz
Caída de tensión	<= 1.5 V (estado cerrado)
Consumo de corriente	<= 35 mA (sin carga)

Primera temporización	< 15 ms
Respuesta del relé	< 2 ms
Recuperación de temporización	< 2 ms
Configuración	Sin ajuste de sensibilidad
Profundidad	50 mm
Altura	50 mm
Anchura	18 mm
Peso del producto	0.19 kg

Entorno

Certificaciones de producto	CE CSA UL
Temperatura ambiente de funcionamiento	-25...55 °C
Temperatura ambiente de almacenamiento	-40...70 °C
Resistencia a las vibraciones	7 gn, amplitud = +/- 1,5 mm (f = 10...55 Hz) de acuerdo con IEC 60068-2-6
Resistencia a los choques	30 gn (cos φ 11 ms) coordinación IEC 60068-2-27
Grado de protección IP	Aislamiento doble IP65 de acuerdo con IEC 60529

Sostenibilidad de la oferta

RoHS (código de fecha: AASS)	Conforme - desde 0841 - Declaración de conformidad de Schneider Electric Declaración de conformidad de Schneider Electric
REACH	La referencia no contiene SVHC La referencia no contiene SVHC

Información Logística

País de Origen	Indonesia
----------------	-----------

Garantía contractual

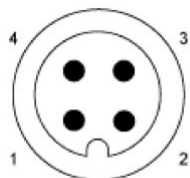
Warranty period	18 months
-----------------	-----------

Hoja de características del producto XUK9APANM12

Conexiones y esquema

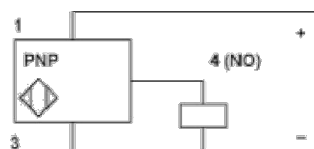
Esquemas de cableado

Conector M12



- 1: (+)
- 3: (-)
- 4: OUT/Salida

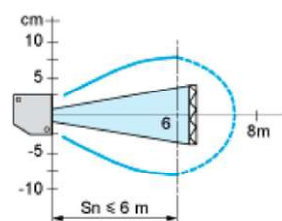
Salidas PNP



Hoja de características del producto XUK9APANM12

Curvas de rendimiento

Curvas de detección



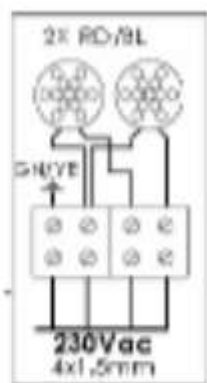
Con reflector XUZC50

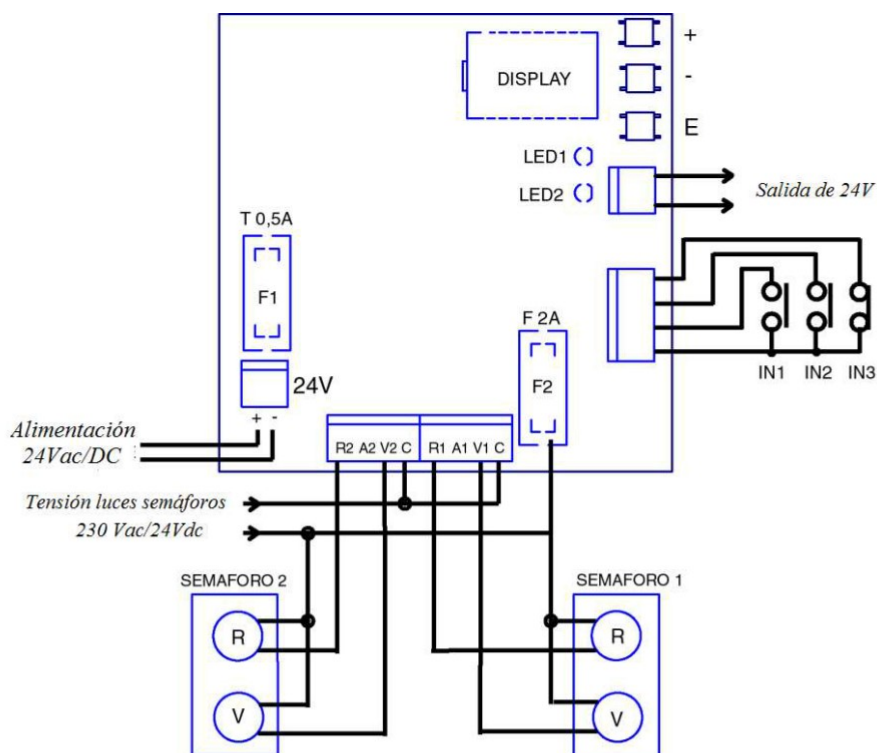
3.9. Ficha de datos del semáforo y la placa de conexión para el semáforo.

CARACTERÍSTICAS

Alimentación	230 V ac
Consumo	2 x 3 W
Horas de vida	100.000
Peso	2,7 kg
Colores disponibles	Rojo y verde
Dimensiones	180 x 290 x 410 mm
Diámetro óptica	Ø120
Grado de protección	IP 65

Esquema del cableado





DESCRIPCIÓN DEL PRODUCTO

La central para semáforo es utilizada para la gestión de semáforos de dos o tres luces en los estacionamientos, residencias y en todos aquellos ambientes en los cuales es necesario regular las entradas.

La central está equipada de interbloques eléctricos y mecánico, para que no suceda en caso de avería el encendido contemporáneo de las luces verdes del semáforo1 y del semáforo2.

La central puede ser utilizada con semáforos de luz a 24V o bien a 230V.

La central se presenta:

- versión básica (alimentación 24V ac/dc corriente alternada o corriente continua)

CONEXIONES ELÉCTRICAS

- 1) + 24Vac/DC alimentación.
- 2) - 24Vac/DC alimentación.
- 3) R2. Salida relé (contacto limpio 5A máx) luz roja semáforo 2.
- 4) A2. Salida relé (contacto limpio 5A máx) luz naranja semáforo 2.
- 5) V2. Salida relé (contacto limpio 5A máx) luz verde semáforo 2.
- 6) Común relé semáforos.
- 7) R1. salida relé (contacto limpio 5A máx) luz roja semáforo 1.
- 8) A1. salida relé (contacto limpio 5A máx) luz naranja semáforo 1.
- 9) V1. salida relé (contacto limpio 5A máx) luz verde semáforo 1.
- 10) Común relé semáforos.
- 11) Común mandos (+24VDC)
- 12) IN1 (Input1), entrada para dispositivos de detección (N.O o N.C) semáforo 1
- 13) IN2 (Input2), entrada para dispositivos de detección (N.O o N.C) semáforo 2
- 14) IN3 (input 3), entrada de emergencia N.C. o N.O.
- 15) Salida tensión auxiliar +24V, 400mA max
- 16) Salida tensión auxiliar -24V, 400mA max

DEFINICIÓN DE LOS COMPONENTES DE LA CENTRALITA

F1- fusible para la protección de la salida auxiliar 24V

F2- fusible para la protección de la salida luces semáforo.

LED1 – LED de tres colores que indica el estado (luz roja/naranja/verde) del semáforo 1.

LED2 – LED de tres colores que indica el estado (luz roja/naranja/verde) del semáforo 2.

DISPLAY- Pantalla de 4 dígitos para los ajustes de funcionamiento

IN3 - La activación de la entrada 3 (EMERGENCIA) en el caso de los semáforos con **dos** implica el inicio de una secuencia de luces intermitentes rojas de los semáforos, hasta la desactivación de l'input3.

La activación de la entrada (EMERGENCIA) en el caso de los semáforos con **tres** luz durante el verde de uno de los dos semáforos causa la activación de la luz naranja, antes de las luces rojas intermitentes ; esto para evitar frenazos bruscos por la persona que procede con la luz verde,

DATOS TÉCNICOS

Alimentación tarjeta: 24V ac/DC (+- 10%).

Consumo de corriente en reposo: 0.02 A (a 24VDC).

Corriente máxima de conmutación relé 5A.

Corriente máxima accesorios : 0.4A a 24Vac/DC.

Alimentación cuadro con transformador: 230V ac (+- 10%) 50Hz.

Consumo de corriente en reposo: 0.04 A (a 230Vac).

Alimentación cuadro con transformador y baterías: 230V ac (+- 10%) 50Hz.

Consumo de corriente en reposo: 0.04 A (a 230Vac).

Capacidad de las baterías de emergencia: 3Ah

Tiempo de carga completa baterías de emergencia: 24h

Pliego de Condiciones



Índice del pliego de condiciones.

1.	Alcance del pliego de condiciones.	111
2.	Condiciones generales.	111
2.1.	Condiciones legales.	111
2.1.1.	Condiciones generales de uso.	111
2.1.2.	Dirección.	112
2.1.3.	Responsabilidad.	112
2.1.4.	Subcontratación.	112
2.1.5.	Conservación de los elementos para la instalación.	112
2.1.6.	Rescisión del contrato.	112
2.2.	Condiciones económicas.	113
2.2.1.	Liquidación en caso de rescisión.	113
2.2.2.	Fianza	113
2.2.3.	Precios de la instalación.	113
2.2.4.	Revisión de los precios.	113
2.2.5.	Retrasos en la ejecución.	113
3.	Condiciones Técnicas.	113
3.1.	Condiciones de fabricación y montaje.	113
3.1.1.	Carga de diseño.	113
3.1.2.	Montaje.	114
3.1.3.	Programa del autómeta.	114
3.2.	Ensayos.	114
3.3.	Mantenimiento.	114

Antes de indicar todas condiciones que pueden influir en la realización de este proyecto cabe indicar que cuando en este documento se está mencionado a la empresa se trata de la empresa encargada de la realización del proyecto, mientras que cuando se menciona al cliente se indica a la empresa que compra este proyecto.

1. Alcance del pliego de condiciones.

En el presente pliego de condiciones se regularán todas las relaciones que existen entre el cliente, el promotor del proyecto y el contratista que lo va a ejecutar. Por este motivo este pliego de condiciones contiene toda la información necesaria para que esas relaciones sean lo más adecuadas posible al tratarse de un documento vinculante.

Por todo lo mencionado anteriormente, en este apartado del proyecto se señalan los derechos, obligaciones y responsabilidades que hay entre las todas las partes que se encargan del proyecto. Las condiciones que indicadas en este documento se encuentra dividido en dos partes claramente diferenciadas: condiciones generales y condiciones técnicas particulares.

La primera parte del pliego de condiciones se recoge fundamentalmente una descripción general del contenido del proyecto, sus características más importantes y los principales aspectos legales, administrativos y económicas. En cuanto a la segunda parte del mismo, contiene el conjunto de normas, instrucciones, especificaciones técnicas y de seguridad que deben tenerse en cuenta a la hora de realizar el proyecto.

2. Condiciones generales.

2.1. Condiciones legales.

Todas las unidades de obra se ejecutarán cumpliendo las prescripciones indicadas en los Reglamentos de Seguridad y Normas Técnicas de obligado cumplimiento para este tipo de instalaciones, tanto en el ámbito internacional por su posible empleo en el extranjero.

2.1.1. Condiciones generales de uso.

Este proyecto está diseñado para controlar la entrada de vehículos en una empresa, en general está diseñado para camiones de 3 ejes o cabezas tractoras con remolque de 2 ejes, aunque se puede emplear el control de otros vehículos siempre que no supere el peso máximo que admite la báscula diseñada.

La instalación de este proyecto se debe de realizar la más próximo a la entrada de los camiones con carga en la empresa cliente y su uso será para el explicado anteriormente. En cualquier otro uso de esta instalación la empresa no se hace responsable de los posibles efectos sobre las personas, elementos que se encuentren en la instalación o sobre la próxima instalación.

2.1.2. Dirección.

El ingeniero técnico director debe de ser una persona capaz de interpretar y hacer ejecutar correctamente el contenido del proyecto, pudiendo ser el mismo ingeniero técnico proyectista. El ingeniero técnico director de la obra resolverá cualquier cuestión que surja referente a la calidad de los materiales empleados, además, de todos los problemas que se planteen durante la ejecución de los trabajos referentes al presente proyecto.

2.1.3. Responsabilidad.

Tanto la empresa como el cliente deben de conocer y aceptar las responsabilidades asignables bajo las condiciones establecidas.

El ingeniero técnico proyectista queda responsable de todos los daños que pudieran darse de un mal diseño, pero no será responsable de los elementos y materiales que se han comprado. En definitiva, el proyectista será el responsable de los problemas que provoque el programa realizado, además del montaje de todos los elementos del proyecto.

2.1.4. Subcontratación.

Ninguna parte del proyecto podrá ser subcontratada sin el consentimiento previo del ingeniero técnico director. Las solicitudes para ceder cualquier parte del contrato deberán formularse por escrito y acompañarse con un documento que acredite su capacidad y profesionalidad de los trabajos a realizar y la calidad de los mismos por la parte de la empresa. La aceptación del subcontrato no relevará a la empresa encargada de la ejecución del proyecto de su responsabilidad contractual.

2.1.5. Conservación de los elementos para la instalación.

La conservación de los elementos entregados al cliente para la ejecución del proyecto, así como su traslado desde la empresa correrán por parte de la empresa. Realizando esta la comprobación de los productos a la llegada al punto de entrega.

2.1.6. Rescisión del contrato.

Se consideran causas suficientes para la rescisión del contrato las siguientes causas:

- PRIMERA: Muerte o incapacitación de alguna de las partes.
- SEGUNDA: Quiebra de la empresa.
- TERCERA: Modificaciones del proyecto en más de un 20% del formato inicial.
- CUARTA: EL no comienzo en el plazo estipulado.
- QUITA: Incumplimiento de las condiciones estipuladas en el contrato en mala fe.
- SEXTA: Finalización del proyecto sin que se cumpla las exigencias iniciales.
- SÉPTIMA: Abandono del proyecto sin causas justificada.

2.2. Condiciones económicas.

2.2.1. Liquidación en caso de rescisión.

Siempre que se quiera rescindir el contrato por causas mencionadas en las condiciones administrativas o bien tras un acuerdo entre ambas partes se le abonará a la empresa el proyecto y el material adquirirlo para el desarrollo.

Cuando se rescinda el contrato, esto llevará implícito la retención de la fianza para los gastos de diseño y derivados del trabajo realizado hasta el momento.

2.2.2. Fianza

En el contrato se deberá reflejar la fianza que la empresa deberá disponer como garantía de cumplimiento del propio contrato. De no estipularse la fianza en el contrato, se entiende que se adoptará como garantía un 35% del valor final del proyecto.

2.2.3. Precios de la instalación.

La empresa presentará al formalizarse el contrato la relación de los precios de las unidades de la instalación y desarrollo que comprende el proyecto, además de los salarios correspondientes.

2.2.4. Revisión de los precios.

En el contrato se acordará si el cliente tiene derecho a la revisión de precios y de forma de aplicarlos. Esto se aplicará en base a la opinión del ingeniero técnico director según los criterios oficiales.

2.2.5. Retrasos en la ejecución.

Si hay un retraso en la ejecución del proyecto pasada la semana de margen establecida, la empresa se hará cargo de las pérdidas que le genere al cliente no disponer de la instalación en el tiempo establecido.

3. Condiciones Técnicas.

3.1. Condiciones de fabricación y montaje.

3.1.1. Carga de diseño.

La báscula ha sido diseñada para soportar una masa máxima autorizada (MMA) de 36 toneladas, ya que en esta báscula pueden subir tanto camiones rígidos o cabeza tractora con remolque como ya se ha comentado anteriormente. Por lo tanto, cualquier fallo, avería o incidente provocado por subir a la báscula un camión cuyo peso sea superior al dicho anteriormente exhumará de responsabilidad al ingeniero proyectista o a la empresa encargada del proyecto y todos los problemas causados deberán resolverse con la responsabilidad de la persona encargada permitir el paso del camión a la báscula.

3.1.2. Montaje.

La instalación de los elementos necesarios para el correcto funcionamiento de todo el proyecto se efectuará conforme a lo establecido en los diferentes apartados recogidos a lo largo de todo el proyecto, cuya interpretación siempre quedará supeditada a la correcta interpretación del ingeniero técnico director.

3.1.3. Programa del autómatas.

El programa realizado para controlar el proyecto estará efectuado conforme lo indicado en los diferentes apartados que se han descrito dentro de todo este proyecto para que haya un correcto funcionamiento de toda la instalación. La interpretación de todos los documentos explicados dentro de este proyecto quedará supeditada a la correcta interpretación del ingeniero técnico director.

3.2. Ensayos.

Antes de la puesta en marcha del proyecto, la empresa encargada del proyecto debe de realizar una serie de comprobaciones, en función de los criterios del técnico director de obra, a todos los aparatos y cableado que se ha instalado correctamente de acuerdo con las normas establecidas con unas condiciones satisfactorias para su uso.

Todas las pruebas realizadas serán presenciadas por el ingeniero técnico director de obra. Los resultados de los ensayos serán pasados en certificados indicando la fecha de realización y nombre de la persona a cargo de dicho ensayo, así como el puesto que posee.

3.3. Mantenimiento.

En este apartado se detallan los pasos para realizar el mantenimiento necesario una vez estén instaladas todas las partes del proyecto. Un mantenimiento preventivo debe ser necesario para un correcto funcionamiento y buen estado de todos los elementos.

Para ello, el encargado de dicha acción puede ser el cliente mediante sus propios medios siguiendo las instrucciones o mediante la contratación de un tercero que sea profesional.

Este proyecto ha sido diseñado para evitar cualquier mantenimiento excesivo de los elementos, por lo tanto, las celdas de carga de la báscula junto con el motor de la barrera automática serán los elementos que precisarán un mayor cuidado.

Las celdas de cargas empleadas para la construcción de la báscula deben de seguir el mantenimiento como se explica en los anexos de los elementos de los proyectos relacionados con dichas celdas, además, hay que revisar el peso que se obtiene de cada una de las celdas y su puesta a cero como se indica en los anexos anteriores ya mencionados.

Otro de los elementos del proyecto que necesita de un control es el motor de apertura de la barrera automática ya que, aunque se han seleccionado una barrera que posee un motor con buenas características y por lo tanto no requiere un mantenimiento excesivo excepto que si el cliente note un desgaste o un mal funcionamiento de dicho motor.

En cuanto a lazo inductivo el cual está formado por el detector y el cable soterrado en el suelo, el mantenimiento a llevar a cabo es igual de importante que la calidad del mismo. Mediante un mantenimiento preventivo realizando una inspección interna para detectar el desgaste del cable o rotura de este sustituyendo el cableado en mal estado o previniendo un cambio futuro. Se podrá evitar un mantenimiento correctivo con la habitual reparación urgente tras una avería que obligará a detener todo el proceso.

Presupuesto



Índice del presupuesto.

1. Introducción.....	121
2. Presupuesto de las piezas.	121
3. Presupuesto mano de obra.	121
4. Presupuesto final.....	122

1. Introducción.

En este apartado se van a describir el presupuesto total del proyecto teniendo en cuenta las piezas que se emplean en las diferentes partes del proyecto y además de las características dispuestas en el pliego de condiciones.

En este presupuesto se ejecuta en base del coste de cada una de las piezas compradas directamente, la creación del programa de automatización por parte el ingeniero. El objetivo de este apartado del proyecto es determinar el coste de este proyecto teniendo en cuenta los gastos de compra, automatización y el beneficio industrial.

2. Presupuesto de las piezas.

En este primer apartado, se realiza el presupuesto general de las piezas compradas. El precio por unidad de estas piezas será obtenido del proveedor correspondiente para cada una de las piezas.

Pieza	Cantidad	Precio/unidad (€/unidad)	Precio (€)
RevPi Core 3+ 16GB	1	259,84	259,84
RevPi DIO	1	177,48	177,48
USB - RS485 Converter	1	7,54	7,54
CODESYS Control for Raspberry Pi MC SL	1	75	75
Célula de carga CBL	4	547,52	2190,08
Placa conexión inteligente CLM8	1	168	168
CASTLGUA Laumas Elettronica Caja protección CLM8	1	56	56
Cámara DS 2CD7A26G0/P IZ(H)S HIKVISION	1	956,41	956,41
Barrera Park 30 XT Aprimatic	1	1333,42	1333,42
Fotocélula REFLEX XUK9APANM12	2	62,6	125,2
Reflector XUZC50	2	11,76	23,52
Sensor inductivo LDD1PA2DU24	1	110,3	110,3
Switch PoE GTS-A1-06-42 6 PUERTOS	1	42,43	42,43
Tablet PIPO X4 64GB Intel Cherry Trail T3 Z8350	1	374,77	374,77

Tabla 3. Tabla de los elementos empleados en el proyecto y su coste.

3. Presupuesto mano de obra.

Otro apartado a considerar dentro del presupuesto de este proyecto es el tiempo que ha necesitado el proyectista en poder realizar todos los aspectos del programa entre los que destacan el tiempo dedicado a la búsqueda de información y de los elementos hardware que conforman el proyecto y también el empleado para la realización del programa.

En este apartado se indicarán las horas totales que se han empleada para poder realizar todo el proyecto y el precio por hora de trabajo del ingeniero.

Proceso	Horas	Precio/Hora (€/h)	Total (€)
Familiarización con los programas empleados	35	30	1050
Selección del material	30	30	90
Realización del programa	70	30	2100

Tabla 4. Tabla de representación del tiempo empleado y su coste.

4. Presupuesto final.

El último apartado del presupuesto se obtendrá la suma de ambos apartados anteriores con lo cual obtenemos el precio de ejecución del material (PEM).

Además, a este precio obtenido también se le sumará un porcentaje de corresponderá a los gastos generales (20%) y otro porcentaje de beneficio industrial (6%). La suma de todas estas cantidades mencionadas anteriormente corresponderá al precio de ejecución por contrata (PEC).

El último proceso que se realizará en este apartado ya será la suma del 21% correspondiente al IVA al precio obtenido anteriormente, lo cual nos hará obtener el precio total de este proyecto.

Proceso	Precio (€)
Presupuesto total de las piezas	5899,99
Presupuesto total mano de obra	4050
Presupuesto de ejecución del material (PEM)	9949,99
Gastos generales (20% del PEM)	1990,00
Beneficio industrial (6% del PEM)	597,00
Presupuesto de ejecución por contrata (PEC)	12536,99
Honorarios por realización del proyecto (6% del PEC)	626,85
TOTAL SIN IVA	13163,84
IVA (21% del total sin IVA)	2764,41
TOTAL	15928.24

Tabla 5. Tabla que muestra el coste final del proyecto.

El precio total al que asciende este proyecto es de QUINCE MIL NOVECIENTOS VEINTIOCHO CON VEINTICUATRO EUROS.